

**UNIVERSITY OF OSLO**  
**Department of Informatics**

**An in depth  
examination of semi  
floating gate ultra  
low voltage flip-flops  
for high speed  
applications**

Master thesis

Erik J. Simenstad,  
Erlend J. S. Johansen

1. August 2013





# An in depth examination of semi floating gate ultra low voltage flip-flops for high speed applications

Erik J. Simenstad, Erlend J. S. Johansen

1. August 2013





# Abstract

In this thesis 4 different ultra low voltage (ULV) flip-flops are presented. Floating gates has been exploited to significantly increase the drain-source current. This technique has proved to decrease the delay significantly and shown that these flip-flops can perform at high speed operations for near subthreshold voltages ( $300mV$ ).

The ULV flip-flops proved to be faster, with a delay up to 20 times faster, than other flip-flop topologies presented in this thesis. The ULV flip-flops also proved to have very little setup and hold times.

With regards to yield, the ULV flip-flops proved to be better at higher frequencies, above  $1MHz$ , than the other flip-flop topologies. One of the ULV flip-flops outperformed the others with a much better yield at all frequencies and supply voltages.

In terms of EDP the ULV flip-flops revealed very good properties. Overall the ULV flips-flops had significantly better EDP, at all frequencies and all supply voltages, than the comparison flip-flop.

One of the flip-flops has been selected for layout design. For the layout the floating gates have been implemented by using MIM capacitors to create crosstalk between two metal strips in the same layer. The layout design uses floating bulks by implementing deep n-wells and enclose the p-substrate for all nMOS transistors with n-wells. This way the p-substrates of all the nMOS transistors have been separated.

The simulations and layout presented in this thesis have been performed in Cadence TSMC  $90nm$  CMOS process.



# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Motivation . . . . .	1
1.2	Previous work . . . . .	2
1.3	Thesis Outline . . . . .	2
<b>2</b>	<b>Background</b>	<b>3</b>
2.1	Low power and low voltage design . . . . .	3
2.2	Power consumption . . . . .	4
2.3	Floating and semi floating gates . . . . .	5
2.4	Mismatch and variation . . . . .	6
2.5	Flip-flops . . . . .	6
<b>3</b>	<b>Ultra Low Voltage Logic (ULV Logic)</b>	<b>9</b>
3.1	Sizing . . . . .	9
3.2	Bulk bias considerations . . . . .	11
3.3	Transistor config. for strength values in TSMC 90nm . . . . .	12
3.4	ULV using semi floating gates . . . . .	12
3.4.1	The boosted QM signal . . . . .	13
3.4.2	Input capacitance considerations . . . . .	14
<b>4</b>	<b>Flip-flops</b>	<b>17</b>
4.1	Key parameters and definitions . . . . .	17
4.2	Standard flip-flop designs . . . . .	19
4.2.1	Conventional Flip-Flop . . . . .	19
4.2.2	Nikolic flip-flops . . . . .	20
4.2.3	Comparison . . . . .	23
4.3	ULV flip-flops . . . . .	23
4.3.1	Basic ULV flip-flop (FFA) . . . . .	24
4.3.2	Input considerations for simulations . . . . .	28
4.3.3	Flip-Flop inverters . . . . .	30
4.3.4	The evaluate transistor . . . . .	32
4.3.5	Flip-flop B (FFB) . . . . .	33
4.3.6	Flip-flop C (FFC) . . . . .	36
4.3.7	Flip-flop D (FFD) . . . . .	38
4.3.8	Flip-flop E (FFE) . . . . .	39
4.3.9	Summary table . . . . .	41
4.4	Flip-flops cascaded in shift register . . . . .	41

<b>5</b>	<b>Robustness, mismatch and variations</b>	<b>45</b>
5.1	Frequency yield . . . . .	46
5.2	Voltage scaling yield . . . . .	48
5.3	Delay variation . . . . .	49
<b>6</b>	<b>Power and energy considerations</b>	<b>53</b>
6.1	Power considerations at different frequencies . . . . .	54
6.1.1	Static power . . . . .	54
6.1.2	Dynamic power, delay and EDP . . . . .	57
6.2	Power considerations at different supply voltages . . . . .	59
6.2.1	Static power . . . . .	60
6.2.2	Dynamic power . . . . .	62
6.2.3	Delay . . . . .	64
6.2.4	Energy Delay Product . . . . .	65
<b>7</b>	<b>Comparison between the ULV flip-flops</b>	<b>67</b>
7.1	FFB . . . . .	67
7.2	FFC . . . . .	68
7.3	FFD . . . . .	68
7.4	FFE . . . . .	68
<b>8</b>	<b>Layout and chip design</b>	<b>71</b>
8.1	Layout . . . . .	71
8.1.1	Deep n-well implentation for floating bulk . . . . .	71
8.1.2	Crosstalk as floating gate capacitance . . . . .	72
8.1.3	Final flip-flop layout . . . . .	75
8.1.4	Amplifier . . . . .	77
8.1.5	Post-layout simulations . . . . .	79
8.2	PCB and measurements . . . . .	81
<b>9</b>	<b>Conclusion</b>	<b>85</b>
9.1	Further work . . . . .	86
<b>A</b>	<b>Comparison reference sheet</b>	<b>87</b>
<b>B</b>	<b>Comments on simulations</b>	<b>89</b>
<b>C</b>	<b>Ocean script for simulations</b>	<b>91</b>
C.1	Yield simulations . . . . .	91
C.2	Power simulations . . . . .	92
<b>D</b>	<b>MATLAB code for simulations</b>	<b>97</b>
D.1	Yield simulations . . . . .	97
D.2	Power simulations . . . . .	102
<b>E</b>	<b>Additional layout close up</b>	<b>105</b>

# List of Figures

2.1	Moore's law . . . . .	4
2.2	Floating gate . . . . .	5
2.3	Latch versus flip-flop . . . . .	7
3.1	Normalized transistor strength . . . . .	9
3.2	Imbalance factor . . . . .	10
3.3	Bulk bias . . . . .	11
3.4	Semi floating gate . . . . .	13
3.5	Precharge for semi floating gate . . . . .	13
3.6	Evaluate for semi floating gate . . . . .	14
3.7	Input capacitance . . . . .	14
3.8	Input load reduction . . . . .	15
4.1	Signal timing definitions . . . . .	18
4.2	Schematic: CTGFF . . . . .	20
4.3	Schematic: TGFF . . . . .	21
4.4	Schematic: Sdff . . . . .	21
4.5	Schematic: MSaff . . . . .	22
4.6	Schematic: basic flip-flop(non-differential) . . . . .	24
4.7	Schematic: differential basic flip-flop . . . . .	25
4.8	Simulation: basic flip-flop, 50MHz . . . . .	25
4.9	Simulation: basic flip-flop, 12.5MHz . . . . .	26
4.10	Simulation: basic differential flip-flop, 50MHz . . . . .	26
4.11	Simulation: basic differential flip-flop, 12.5MHz . . . . .	27
4.12	Simulation: basic differential flip-flop, 3.125MHz . . . . .	28
4.13	Recharge transistors . . . . .	29
4.14	Flip-flop inverters . . . . .	30
4.15	Simulation: Inverter strength issue . . . . .	31
4.16	Evaluate transistors . . . . .	32
4.17	Schematic: FFB . . . . .	33
4.18	Simulation: FFB . . . . .	35
4.19	Schematic: FFC . . . . .	36
4.20	Schematic: FFD . . . . .	38
4.21	Schematic: FFE . . . . .	39
4.22	Schematic: shift register . . . . .	41
4.23	Clock leakage in shift registers . . . . .	42
5.1	Illustration of yield . . . . .	45

5.2	Simulation: Frequency yield . . . . .	46
5.3	Simulation: Frequency yield (short input time) . . . . .	47
5.4	Simulation: Voltage scaling yield . . . . .	48
5.5	Delay variation FFB . . . . .	49
5.6	Delay variation FFC . . . . .	50
5.7	Delay variation FFD . . . . .	50
5.8	Delay variation FFE . . . . .	51
6.1	Simulation: Frequency sweep - static power . . . . .	54
6.2	Simulation: Frequency sweep - normalized static power . . . . .	55
6.3	Simulation: Frequency sweep - norm. static power (no FFC) . . . . .	56
6.4	Simulation: Frequency sweep - dynamic power . . . . .	57
6.5	Simulation: Frequency sweep - delay . . . . .	58
6.6	Simulation: Frequency sweep - EDP . . . . .	58
6.7	Simulation: $V_{dd}$ sweep - static power . . . . .	60
6.8	Simulation: $V_{dd}$ sweep - normalized static power . . . . .	61
6.9	Simulation: $V_{dd}$ sweep - dynamic power . . . . .	62
6.10	Simulation: $V_{dd}$ sweep - normalized dynamic power . . . . .	62
6.11	Simulation: $V_{dd}$ sweep - dynamic vs static power . . . . .	63
6.12	Simulation: $V_{dd}$ sweep - delay . . . . .	64
6.13	Simulation: $V_{dd}$ sweep - normalized delay . . . . .	65
6.14	Simulation: $V_{dd}$ sweep - EDP . . . . .	65
6.15	Simulation: $V_{dd}$ sweep - normalized EDP . . . . .	66
8.1	Deep n-well cross section . . . . .	71
8.2	Layout: Deep n-well . . . . .	72
8.3	Capacitor: Transistor as lumped RC . . . . .	73
8.4	Layout: MIM capacitor . . . . .	74
8.5	Layout: Flip-flop . . . . .	76
8.6	Schematic: Amplifier . . . . .	77
8.7	Layout: Amplifier . . . . .	78
8.8	Simulation: Layout, 1MHz . . . . .	79
8.9	Simulation: Amplifier . . . . .	80
8.10	Simulation: Amplifier (close up) . . . . .	80
8.11	Layout: PCB . . . . .	81
8.12	PCB 1 . . . . .	82
8.13	PCB 2 . . . . .	82
8.14	Measurement equipment . . . . .	83
E.1	Layout: Flip-flop close up . . . . .	105
E.2	Layout: Flip-flop close up inverters . . . . .	106
E.3	Layout: Flip-flop close up evaluate . . . . .	107
E.4	Layout: Amplifier close up left . . . . .	108
E.5	Layout: Amplifier close up center . . . . .	109
E.6	Layout: Amplifier close up right . . . . .	110

# List of Tables

3.1	Transistor strength . . . . .	12
4.1	Comparison: CTGFF, TGFF, SDFF, MSAFF . . . . .	23
4.2	Recharge transistor strength . . . . .	29
4.3	Evaluate vs inverter strength . . . . .	32
4.4	FFB . . . . .	35
4.5	FFC . . . . .	36
4.6	FFD . . . . .	38
4.7	FFE . . . . .	40
4.8	ULV flip-flops: Summary . . . . .	41
7.1	ULV flip-flop comparison . . . . .	67
8.1	Schematic vs layout . . . . .	79





# Acronyms

<i>AC</i>	Alternating current
$\beta$	Current factor
<i>CMOS</i>	Complementary metal oxide semiconductor
<i>CPU</i>	Central processing unit
<i>CTGFF</i>	Conventional transmission-gate flip-flop
<i>EDP</i>	Energy delay product
<i>E<sub>n/p</sub></i>	Evaluate transistor
<i>FF</i>	flip-flop
<i>gnd</i>	Ground signal
<i>IF</i>	Imbalance factor
<i>I<sub>off</sub></i>	Off current
<i>I<sub>on</sub></i>	On current
<i>I<sub>rb</sub></i>	Reverse bias current
<i>I<sub>ST</sub></i>	Subthreshold current
<i>K<sub>n/p</sub></i>	Keeper transistor
<i>L</i>	Length
<i>MIM</i>	Metal-insulator-metal
<i>MOSFET</i>	Metal oxide semiconductor field-effect transistor
<i>MSAFF</i>	Modified sense amplifier based flip-flop
<i>nMOS</i>	N-channel MOSFET
<i>PCB</i>	Printed circuit board
<i>pMOS</i>	P-channel MOSFET
<i>R<sub>n/p</sub></i>	Recharge transistor
<i>SDFF</i>	Semi-dynamic flip-flop
<i>SRAM</i>	Static random-access memory
<i>t<sub>c-q</sub></i>	Clock to output delay
<i>t<sub>d-q</sub></i>	Input to output delay
<i>TGFF</i>	Transmission-gate flip-flop
<i>t<sub>hold</sub></i>	Hold time
<i>t<sub>input</sub></i>	Input time
<i>t<sub>max</sub></i>	Maximum input time
<i>t<sub>setup</sub></i>	Setup time

<i>TSMC</i>	Taiwan Semiconductor Manufacturing Company
<i>ULV</i>	Ultra-low voltage
<i>UV</i>	Ultraviolet
$V_{dd}$	Supply voltage
$V_{DS}$	Drain-source voltage
$V_{GS}$	Gate-source voltage
$V_T$	Thermal voltage
$V_{TH}$	Threshold voltage
$W$	Width

# Preface

This Master thesis was carried out as part of the degree in *Master of Science* in Nano- and Microelectronics, at the Department of Informatics, Faculty of Mathematics and Natural Sciences, University of Oslo (UiO). The project was initiated in January 2012, and concluded the following year in August 2013. The thesis contributes 60 credits.

Working on this thesis has been a great experience. It has been challenging, inspiring, interesting and rewarding, and it has gained us a deeper understanding of modern day electronics and the challenges that it faces. This experience has granted us a thorough understanding of ULP and ULV design principles and challenges, and has also provided us with the rare opportunity of designing our own chip that has been produced. Finally as a team of two students we have learned a lot about teamwork and forged a lifelong friendship.

First and foremost we would like to express gratitude to our supervisor, Professor Yngvar Berg, for giving us guidance and freedom to explore the technology that interests us. Thank you for all the invaluable feedback that you have given us, and for inspiring us to explore new ideas and try out things that may not work. A special thanks to Olav Stanly Kyrvestad for helping with layout, chip and PCB design, as well as setting up the lab and measurement equipment, and for answering the many technical questions related to these. Also thanks to Amir Hasanbegovic for lending us advice from his experience in chip and layout design.

Many thanks to all our costudents at the master lab, Øystein, Dag, Alexander, Halfdan, Abdul, Thomas, Tor, Martin, and an additional thanks to Patrick and Musa for their help with  $\LaTeX$ . Thanks for providing a great environment, for jokes and laughs, and thanks for many technical discussions as well as casual conversations that makes it easy to keep on going late in the evening.

Lastly we would like to thank our families for supporting us throughout our five years of studies, for motivating us and for believing in us.



# Chapter 1

## Introduction

For many years there has been an increase in the demand for mobile devices and sensor devices. With these devices comes a need for long battery life, small size and high robustness[1]. One of the most effective ways to scale down the power consumption and thus increasing battery life, is by lowering the supply voltage[2]. However this comes at a significant cost on the speed of which the devices can operate. Several solutions to this problem have been presented that can allow for higher operational speed at low supply voltages[3][4][5].

### 1.1 Motivation

Most novel flip-flop designs today focus on low power[6][7][8][9], and does not directly focus on low voltage although low voltage is a natural ending point when designing for low power. They also usually do not focus on high speed operation since low power design usually means sacrificing speed for lower energy consumption.

High speed ULV flip-flops[10] has been proposed that significantly reduces the delay time of low voltage flip-flops. ULV flip-flops thus presents an exciting way to implement flip-flops for low voltage and low power purposes. With increased operational speed these flip-flops can be combined with other fast ULV logic styles in order to improve speed in bottleneck areas of the circuit design. Furthermore these flip-flops uses semi-floating gate nodes that can reach potentials higher than the supply voltage. This creates a short burst of high  $I_{on}$  current which helps increase robustness as  $I_{on}/I_{off}$  begins to reach problematicly low values in low voltage circuits.

The goal of this thesis is to show not only that ULV flip-flops are much faster than other flip-flop topologies, but also that these flip-flops can be considered for implementation even if low power considerations and yield are important for the circuit.

## 1.2 Previous work

Ultra-low voltage(ULV) logic using semi floating gates is a relatively new idea, proposed first in [11] building on previous work on floating gate logic[12]. Semi-floating gate logic is much easier to implement in modern CMOS technology because precharge is done using transistors instead of external sources. Many different types of circuit components and different styles have previously been explored using ULV semi-floating gate logic[13][14][15][16][17].

ULV flip-flops using semi floating gates have also been proposed[10]. However there is much work left to be done on investigating and creating a comprehensive analysis of ULV flip-flops. In particular there are no good analysis of yield or power considerations for these flip-flops. Furthermore there is a need for better layout simulations and chip measurements to see how well these kind of flip-flops works in a more realistic setting outside of a schematical environment.

## 1.3 Thesis Outline

- Chapter 1 gives an introduction to challenges with regards to ultra low voltage. Motivation for the thesis is explained. Previous work is also explained in this chapter.
- Chapter 2 provides a general explanation of different topics that are useful for understanding the thesis with regards to low power/voltage, floating gates, mismatch and flip-flops.
- Chapter 3 explains how size and bulk bias of the transistors affects the transistor strength. Furthermore it explains the use of semi floating gates and input capacitance.
- Chapter 4 introduces the different flip-flops that will be examined throughout this thesis, including the ULV flip-flops and the flip-flops used for comparison. Results regarding delay, setup time and hold time are also presented for each flip-flop.
- Chapter 5 shows how yield is affected by frequency and voltage scaling by running Monte Carlo simulations. It also shows how the delay varies with process and mismatch variations.
- Chapter 6 gives a thorough examination of how frequency and voltage scaling affects power consumption and EDP. Static power and dynamic power has been examined separately.
- Chapter 7 compares the ULV flip-flops with pros and cons.
- Chapter 8 introduces layout and chip design. Furthermore it explains the considerations that has to be taken for the layout design.
- Chapter 9 concludes all the topics that has been examined throughout the thesis. It also includes a list of future investigations.

## Chapter 2

# Background

### 2.1 Low power and low voltage design

According to Moore's law[18] the transistor count on integrated circuits will double every 18 months as a result of the extremely rapid evolution of electronic technology and design techniques. This is only an empirical number that have held up for many decades as seen in Figure 2.1. However as transistors become smaller and smaller a physical limit is reached that will prevent this trend from continuing forever. As transistors become smaller and smaller, the supply voltage drops naturally with it as well, however the drop in supply voltage is slowed down by the need for high speed operations.

The high demand for portable devices and sensors are the driving forces behind the need for low power and low voltage circuit design. Because battery technology has evolved much slower than CMOS technology[19] the limitations set by the battery needs to be dealt with by designing more energy efficient circuits.

As the need for low power and low voltage devices becomes one of the main driving forces in electronic design[2], the subthreshold and near threshold operation of transistors becomes the dominant operational modes in CMOS logic. In older technologies the current in these modes are often considered a part of the leakage current only, and is logically equal to 0 or *low*. However with low voltage design in mind these currents becomes the main  $I_{on}$  currents present in the circuits and accordingly has to be accounted for.

## Microprocessor Transistor Counts 1971-2011 & Moore's Law

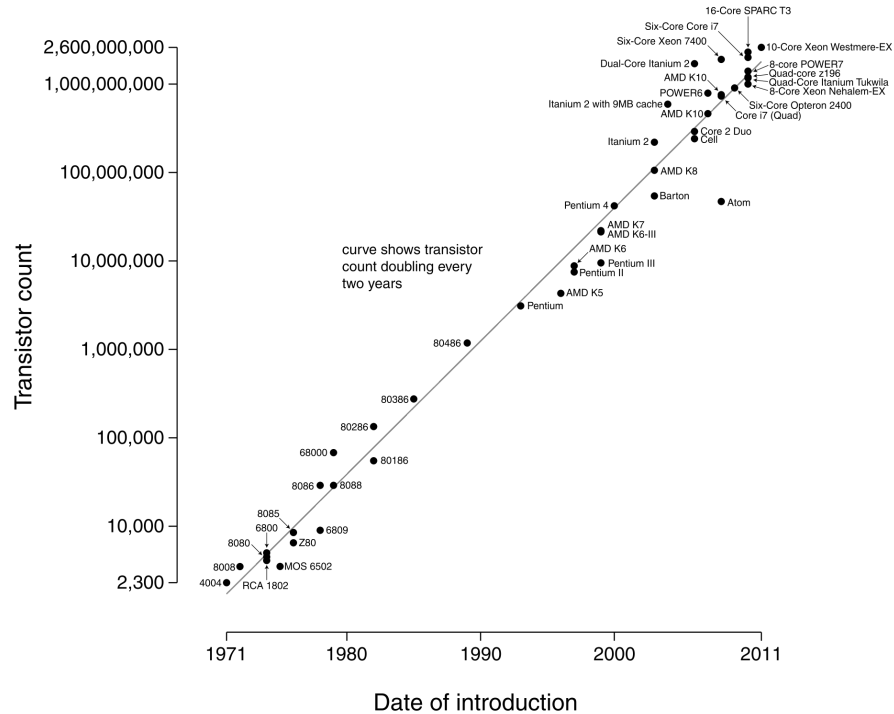


Figure 2.1: Transistor count as predicted by Moore's law shown by the line, with actual CPU transistor counts included to show that it holds up. Source: [20]

By dropping the supply voltage all the way down to subthreshold levels the power consumption can be dramatically reduced[19]. The minimum supply voltage that can be used at perfect balance is about  $2V_T$ [21]. The main problems with working at voltages as low as these is that the negative effects of mismatch and variation has much more impact at subthreshold currents. By instead working at near subthreshold voltage one can get better yield and more reliable circuitry while at the same time benefit from low power operation.

## 2.2 Power consumption

With low power and energy saving in focus one of the most important factors to consider is the power consumption of the transistors. In low voltage environments the leakage currents can be a significant contributor to power consumption in circuits which can cause a lot of power consumption to happen when there is actually no activity in the circuit components. This means that when designing circuits for low power at low voltage one needs to consider how active the circuit components are going to be, and design



for low dynamic vs low static power consumption depending on the needs.

Another option that can be considered is sleep mode. A circuit can be designed so that it will only be active short periods at a time. This works well for circuits that needs to do some operations with a specific time interval. Most of the time the circuit can be in sleep mode, where for example the supply voltage can be completely cut off in order to eliminate the leakage current thereby lowering the power consumption to near zero. A short period of time the circuit will be turned on in order to perform it's task. This kind of operation can also take great advantage of additional circuit speed as this will allow for a shorter on-period and increase the sleep time, thereby reducing the power consumption.

## 2.3 Floating and semi floating gates

A floating node is a node in a circuit that is separated from other current paths so that virtually no current can flow out of or into the node. A floating gate node can be made by connecting the gate of a transistor to an input capacitor as seen in Figure 2.2. The gate voltage can then be affected by AC currents, or changes in the voltage value at the other end of the input capacitor. The gate voltage will also be determined by the initial gate voltage which can be precharged by using UV light or several other means of initialization[12].

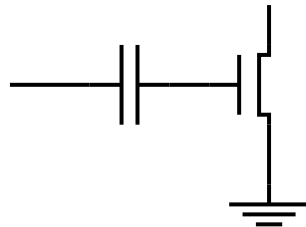


Figure 2.2: A floating gate CMOS transistor created using an input capacitor.

A semi floating gate can be achieved by using a similar setup to that of the floating gate. However instead of precharging the gate node by means of UV-light or other external sources, the precharge is set by a precharge signal usually activated through a recharge transistor. The gate node is then only floating when the precharge signal is cut off. Semi-floating gates have previously been implemented in several different logical gates and other digital components[10][13][14][15][16][17].

At low voltage using modern transistor technology there will be a significant leakage to/from the floating node, so frequent recharge/precharge is needed. Semi floating gates will be discussed in greater detail in section 3.4.

## 2.4 Mismatch and variation

For two identical transistors there are random variations of the difference in the threshold voltage  $V_{TH}$  and current factor  $\beta$ . These factors are dependent on the width and the length of the transistors, given by:

$$\sigma(\Delta V_{TH}) = \frac{A_{V_{TH}}}{\sqrt{WL}}$$
$$\sigma\left(\frac{\Delta\beta}{\beta}\right) = \frac{A_{\beta}}{\sqrt{WL}}$$

Where  $A_{V_{TH}}$  and  $A_{\beta}$  are technology dependent constants.

These are widely accepted models which can be found in [22][23]. In subthreshold region the effects of mismatch is even more significant because of the exponential property of the transistor current. Which means that a small difference in transistor dimensions has a larger impact on the difference in transistor current[24].

The mismatch has a huge impact on  $I_{on}/I_{off}$  [25] because it's reduced by various orders of magnitude for subthreshold voltages compared to above threshold. Consequently this has a huge impact on identical transistors that are connected to the same node as they will suffer from severe robustness degradation[19]. As will be shown in section 3.1 sizing in low voltage applications behaves differently from high voltage and because of this mismatch has to be taken into consideration when designing ULV logic.

## 2.5 Flip-flops

Flip-flops are digital components used to store information. They are usually volatile memory components which means they will not keep their stored value when the power is cut off. Flip-flops in contrast to latches are not transparent during any clock period (see Figure 2.3). The stored bit value within the flip-flop will only change at a specific clock flank at which the flip-flop will latch onto the current value of the input signal. In traditional flip-flop designs this can be achieved by placing two latches in series that are clocked inversely to each other.

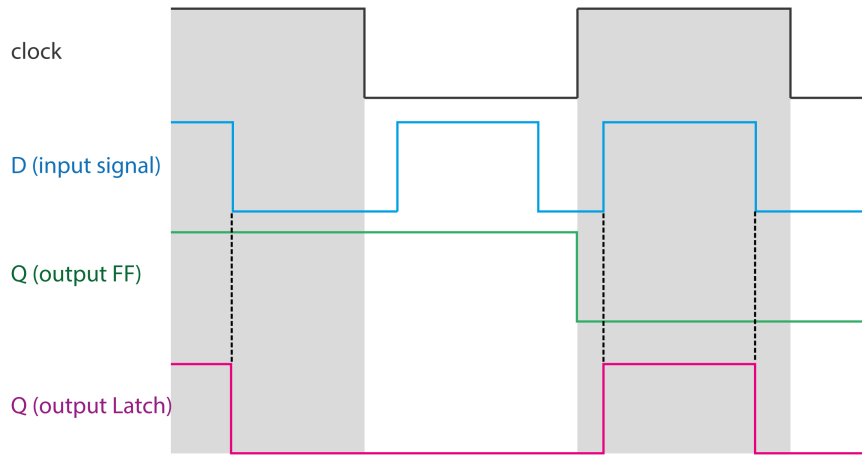


Figure 2.3: A latch will always latch onto the input during a high (or low depending on design) clock period. A flip-flop can only latch onto a signal at the moment of a rising (or falling) clock edge.

The output does not necessarily follow the input signal as seen in Figure 2.3. The output can be inverted, so that at rising clock edge the flip-flop will latch onto the inverted input signal. This will indeed be the case for the ULV flip-flops that are presented in chapter 4.

Flip-flops are commonly used in shift registers for CPU's or other devices that needs fast easy to access memory. SRAM memory is often also designed using flip-flops for memory storage. They are also used for creating state-machines, and can also generally be used to set conditions for clocking of devices and several other uses related to clocking. In short if there is a need for a clocked component capable of storing a bit, a flip-flop is used.

Because flip-flops relies on contention currents between different nodes, the minimum supply voltage tends to be a little bit higher because of the importance of having a big enough difference between  $I_{on}$  and  $I_{off}$  currents[19]. This problem is addressed by the ULV flip-flops presented in [10] which create a much stronger  $I_{on}$  current without increasing the supply voltage. The ULV flip-flops will be the main focus of this thesis, and so will be discussed in great detail in later chapters.

Several other flip-flops have been proposed for low power use[6][7][8][9], notably the flip-flops investigated in [26] which will be further examined in section 4.2.



## Chapter 3

# Ultra Low Voltage Logic (ULV Logic)

### 3.1 Sizing

Transistor sizing at subthreshold voltages works very differently from transistor sizing at higher voltages[27][19][28]. The normalized on current  $I_{on}$  for nMOS and pMOS transistors in the  $90nm$  process that was used for simulations and layout in this thesis is shown as a function of normalized width  $W$  and length  $L$  in Figure 3.1. This result is similar to the results in Alioto [19] where a  $65nm$  process was used.

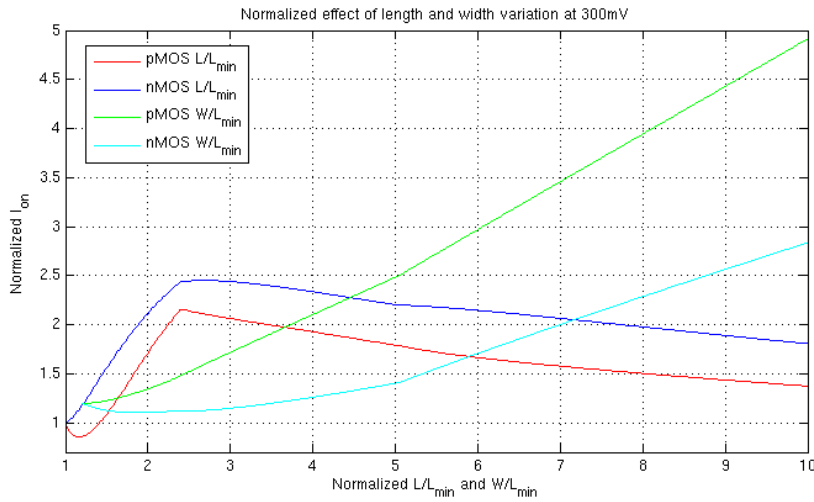


Figure 3.1: Normalized transistor strength at  $300mV$ .

The transistor strength is a relative measure indicating how much more current goes through a transistor compared to the minimum amount based on static configurations of the transistor (size and bulk bias). In Figure 3.1 the minimum transistor current at constant gate voltage represents a transistor strength of 1. When the transistor size is increased enough to double the current, the transistor strength is considered to be doubled as

well.

It can be seen from Figure 3.1 that length is a much better knob for tuning transistor strength than width at low current levels. However if the transistor strength requirement is high (for example in the clock generator) the width will have to be used as well since tuning by length will reach a peak value at about  $240nm - 250nm$  (with  $L_{min} = 100nm$ ).

It might also be desirable to move further up in length to the much “flatter” area located directly after the peak. This will give some headroom for mismatch in length that won’t have a big effect on performance.

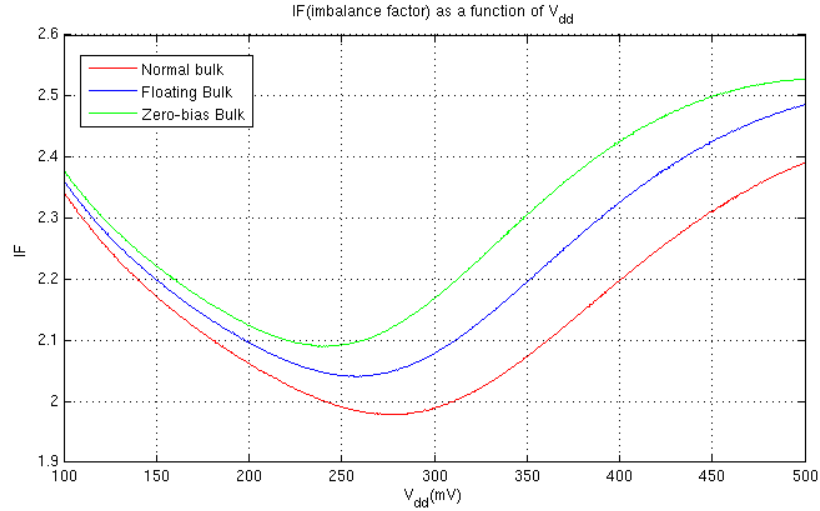


Figure 3.2: Imbalance factor IF at  $\frac{W}{L} = \frac{120nm}{250nm}$

In Alioto [19] the typical imbalance factor(IF) at subthreshold voltage is said to be roughly  $\frac{\beta_n}{\beta_p} = 7$ , however the IF at minimum width  $W = 120nm$  for the  $90nm$  process used in this thesis is roughly  $\frac{\beta_n}{\beta_p} = 2$  as can be seen in Figure 3.2. This value is more typical for above threshold supply voltage and somewhat unexpected for the low supply voltages used. This makes matching transistor strength much simpler, giving more room for tuning the wanted transistor strength levels.

As can be seen in Figure 3.1 the IF is more or less independent of transistor length (the difference between nMOS and pMOS length scaling remains relatively constant). The same is not true for the effect transistor width scaling has on the IF.  $I_{on}$  for the pMOS transistor increases much more rapidly with the transistor width than what it does for the nMOS transistor. Since the nMOS transistor is roughly 2 times as strong as the pMOS transistor even at  $300mV$  for this process, this means that at high values of  $W$  matching pMOS and nMOS transistors will make them very similar in size.

Figure 3.2 also shows a comparison between three different bulk bias possibilities. The first case is “Normal bulk” where nMOS bulk is grounded and pMOS bulk is connected to  $V_{dd}$ . This is the best scenario for mismatch, since this is where the IF is lowest. However the increase in IF is marginal,

and can easily be outweighed by the possible advantages of other bulk biasing methods. The second case is for floating bulk, where leakage currents will set the a floating bulk voltage somewhere between  $V_{dd}$  and  $gnd$ . Finally the third case “zero-bias bulk” is where the bulk bias has been switched between nMOS and pMOS so that nMOS bulk is connected to  $V_{dd}$  and pMOS bulk is connected to  $gnd$ .

### 3.2 Bulk bias considerations

Another effective knob for tuning the transistor strength at ULV is the bulk bias [19]. At subthreshold voltage there is low risk of latch up so the bulk does not necessarily have to be reverse biased which is usually the standard way to bias the bulk.

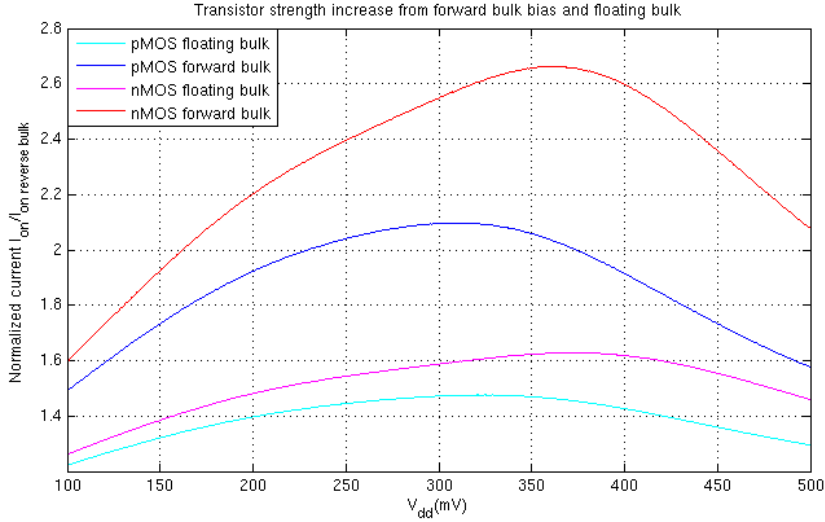


Figure 3.3: Comparison of bulk bias for nMOS and pMOS transistors.

In Figure 3.3 the effect of using different body bias is shown. The normalized transistor current  $I_{on}/I_{rb}$  (where  $I_{rb}$  is the reverse bias current/ordinary body) is equal to the gain in transistor strength compared to the ordinary reverse bias at a given supply voltage. The data was acquired using the 90nm technology used throughout this thesis. One can see that there seems to be a peak of efficiency at about  $V_{dd} = 350mV$ . It can also be seen that for this technology forward and floating body bias is more effective for nMOS transistors than pMOS transistors. This difference seems to be bigger for forward body bias than it is for floating bulk. Furthermore it can be seen that forward body bias gives a much larger transistor strength increase than a floating bulk, which is expected since a floating bulk will settle somewhere between reverse and forward body bias voltage values.

### 3.3 Transistor configuration for specific strength values in Cadence TSMC 90nm

Using the knowledge about bulk-bias and transistor sizing a table can be made for simplification of transistor sizing in TSMC 90nm process that is used throughout this thesis for simulations in Cadence. This is shown in Table 3.1.

Generally if no specifications are made about transistor strength for the circuits, it can be assumed that both nMOS and pMOS transistors have been normalized to a transistor strength of 1. For inverters the inverter strength is the same as the transistor strength that is used for both the nMOS and pMOS transistors for that inverter.

Table 3.1 was generated assuming a supply voltage close to 300mV. Body bias considerations were made before length and width considerations in order to utilize transistors that are as small as possible. Length is increased before width because as previously discussed length is a more powerful scaling factor than width as seen in Figure 3.1. This way the transistor size is kept as small as possible at all times, minimizing area and saving space.

	pMOS			nMOS		
Transistor strength	L(nm)	W(nm)	Bias	L(nm)	W(nm)	Bias
1	100	120	<i>gnd</i>	100	120	<i>gnd</i>
2	220	120	<i>gnd</i>	190	120	<i>gnd</i>
3	250	280	<i>gnd</i>	120	120	$V_{dd}$
4	250	400	<i>gnd</i>	150	120	$V_{dd}$
5	250	560	<i>gnd</i>	190	120	$V_{dd}$
6	250	670	<i>gnd</i>	230	120	$V_{dd}$
7	250	800	<i>gnd</i>	250	470	$V_{dd}$
8	250	900	<i>gnd</i>	250	540	$V_{dd}$
9	250	1020	<i>gnd</i>	250	600	$V_{dd}$

Table 3.1: Transistor size and body-bias that can be used in order to achieve different transistor strength values relative to minimum-size transistors.

### 3.4 ULV using semi floating gates

One of the main principles of the ULV logic is the use of capacitors to increase/decrease the available voltage levels (up to  $3/2V_{dd}$ , down to  $-1/2V_{dd}$ ). The capacitors are charged up in the recharge phase and the saved charge is spent in the evaluate phase. The specifics of what input conditions that allows the capacitors to be charged depends on the function of the circuitry.



### 3.4.1 The boosted QM signal

At ULV the transistor current is largely dominated by the subthreshold current  $I_{ST}$  [19] which is usually given in the simplified form:

$$I \approx I_{ST} = I_0 \frac{W}{L} e^{(V_{GS}-V_{TH})/n \cdot v_t} (1 - e^{-V_{DS}/v_t}) \quad (3.1)$$

Where  $I_0$  is the subthreshold current at  $V_{GS} = V_{TH}$ ,  $V_T$  is the thermal voltage,  $\frac{W}{L}$  is the size ratio and  $n$  is the subthreshold factor.

As has been discussed earlier the relationship between the transistor current and the transistor size is in reality not as simple as  $\frac{W}{L}$  however the transistor current has an approximately proportional relationship to the transistor size  $\frac{W}{L}$ . The gate-source voltage  $V_{GS}$  however has an exponentially proportional effect on the transistor current. It would be very advantageous if  $V_{GS}$  could be boosted to a higher voltage level than  $V_{dd}$  since this would exponentially increase the transistor current, which would make the transistor operate much faster. While this extra current would cause the transistor to use more power which is usually not desirable in ULV logic, it would be advantageous with regards to yield as will be seen in chapter 5.

One way of achieving such a boosted input current signal is by implementing a precharge capacitor at the gate as seen in Figure 3.4. One would get a precharge phase at either a high or low part of the clock period and an evaluate phase in the other half of the clock period (depending on how one connect the clock signal). On the gate side of the capacitor a precharge signal is connected and we name this node *QM*.

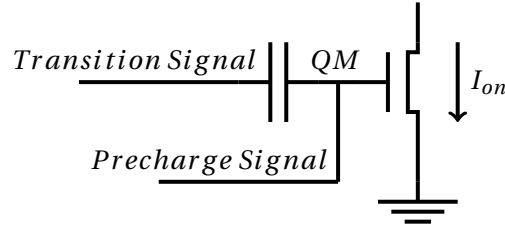


Figure 3.4: A capacitor can be added to the gate of a transistor, allowing energy to be stored and released by changing the transition and precharge signals.

During the precharge phase the *QM* node is charged by the precharge signal to either a high or low value as seen in Figure 3.5.

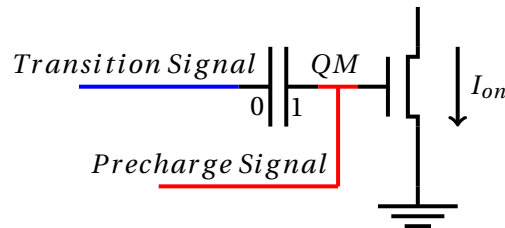


Figure 3.5: *QM* node during precharge phase.

During the Evaluate phase the precharge signal should be disconnected from the  $QM$  node by the use of a transistor. The  $QM$  gate node is now a floating node with a charge. The capacitor has been charged up in the precharge phase, and can be viewed similarly to a battery in series with the gate input as seen in Figure 3.6. When a transition occurs at the transition signal side of the capacitor it will transmit over to the  $QM$  side, however the charge already at the  $QM$  side will help to either make the transition higher or lower in value. Alternatively one can view the capacitor as a battery so that the signal is increased by an extra voltage source in series with the gate signal. This increase is typically  $+\frac{1}{2}V_{DD}$  (for nMOS).

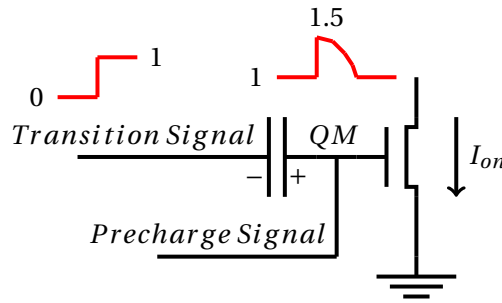


Figure 3.6: QM node during evaluate phase.

For the flip-flops in this thesis the precharge signal will be the input signal  $D$  or  $D'$ , and the transition signal will be the clock signal so that the transition happens just as the evaluate phase starts. This ensures that the flip-flop operation is flank-triggered and does not work like a latch.

### 3.4.2 Input capacitance considerations

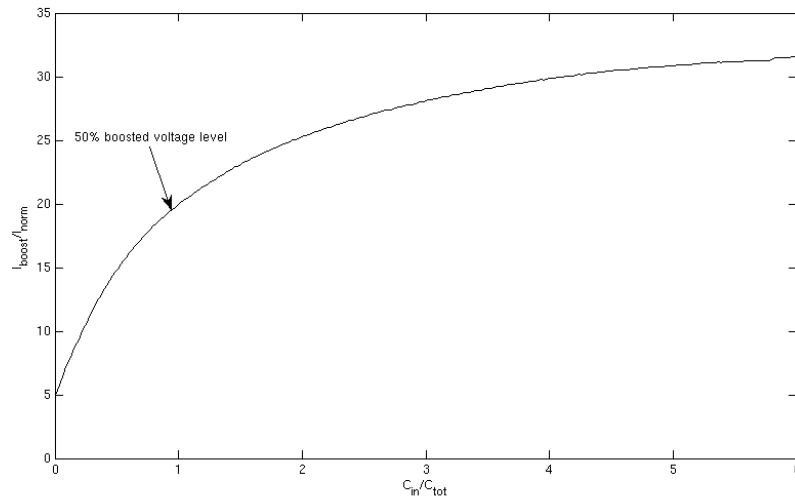


Figure 3.7: Normalized current representing transistor strength, as a function of normalized input capacitance used for current boost at  $300mV$ .

In Figure 3.7 it is clear that there is a very significant gain in transistor on-current when the gate voltage is boosted using an input capacitor. The current gain per capacitance is larger for small capacitors. When the capacitance is high, the current gain is much lower. This is also evident from the formula for capacitive division:

$$\Delta V = \frac{C_{in}}{C_{in} + C_{parasitic}} V_{dd}$$

When the input capacitance is high,  $\Delta V$  will converge towards  $V_{dd}$ . Because higher capacitance requires more area, it is not worth it to use large capacitors.

In the proposed flip-flop design the input signal  $D$  is connected to the  $QM$  node through a source or drain terminal. This means that the load on the  $D$  signal consists of both the gate capacitance of the evaluate transistor and the parasitic capacitances through source and drain of the recharge transistors, as well as the load created by the precharge-capacitor. A keeper transistor will be added later in this thesis, which will contribute more load. Some later flip-flop configurations will also add additional input load.

When the input capacitance increases, this creates a delay for the input signal, making it slower. There is a need for a good balance between input signal speed decay and speed gain caused by the boosted evaluate transistor decreasing the output delay. It turns out that when the boosted voltage level is set at about  $+1/2V_{dd}$  the decreased delay vs the increased delay is at a good balance. For the transistors used in this thesis the input capacitors should be about  $1fF$  in order to achieve this.

In a shift register one flip-flop will have to drive another flip-flop as well as driving an output node. Therefore it is desirable to have as low input load as possible for the flip-flops.

One way to reduce the input load is to avoid connecting the  $D$  signal to the  $QM$  node with a direct path, and instead connect the input signal only to a single gate which in turn drives the node. This way the total amount of load put on the input signal is reduced to only one or two gates. This removes the need for the input signal to drive the critical  $QM$  node. As an example, this can be seen in Figure 3.8.

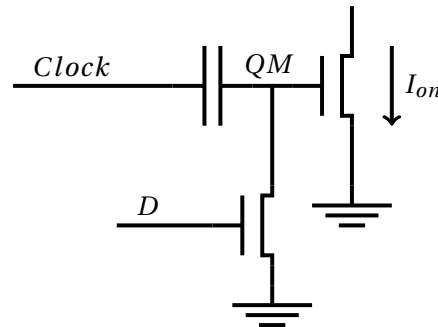


Figure 3.8: A configuration sending the  $D$  signal directly into the gate of a transistor to reduce input load.

The downside of connecting the  $D$  signal in this way is that it adds a small amount of delay since an additional transistor has to switch before the effect of the  $D$  signal reaches the  $QM$  node. This delay is added in the form of longer setup time requirements. In the end the added setup time slows down operation too much and is not worth it.

# Chapter 4

## Flip-flops

### 4.1 Key parameters and definitions

#### Delay

There are several important key parameters when looking at flip-flop design. For speed considerations the most important one is the flip-flop delay. A low delay is advantageous as this will allow for a higher operating frequency of the circuit. When nothing else is stated, the delay is defined as the *clock to output delay*  $t_{c-q}$ ; The time it takes from the clock has reached 50% of  $V_{dd}$  till the output signal has reached the same value. This parameter is very good for comparison between different flip-flops because it does not rely too much on input time considerations which allows for long input times to be used under testing (see input time definition below). Another important parameter is the *input to output delay*  $t_{d-q}$  which is defined in this thesis as the time it takes from the input data signal  $D$  reaches 50% of  $V_{dd}$  till the output signal  $Q$  reaches the same value. This represents the data delay of the flip-flop, and is measured when input time = setup time, and so represents the lowest possible time it takes for the data input to reach the output. Using this definition of delay, it's important that the clock edge is not too uneven.

#### Input time

In this thesis we will define the input time  $t_{input}$  as the time from when the input signal is received at the flip-flop till the rising clock edge when the flip-flop will latch on to this input. This time varies depending on the delay from the previous circuitry and is not a parameter of the flip-flop itself, but rather a description of how the input signal is perceived by the flip-flop.

#### Setup time

The *setup time*  $t_{setup}$  describes the minimum input time that is needed for the flip-flop to properly latch on. It is defined in this thesis as the minimum time from the data input signal reaches 50% till the clock signal reaches the same value with a properly working output that latches on. If the input edge arrives in too late, the flip-flop will

not have enough time to latch on properly and could instead latch on the next rising clock edge or not function properly at all giving unstable output signals.

### Hold time

The flip-flop *hold time*  $t_{hold}$  is similar to the flip-flop setup time. It represents the time that the input signal has to remain stable after the rising clock edge, in order for the flip-flop to hold on to the signal properly. It is defined in this thesis as the minimum amount of time after the output reaches 50% of  $V_{dd}$ , that the input has to remain stable (unchanged) in order for the output to remain stable as well. If the input signal transitions during the hold time of the flip-flop, the output signal could potentially switch too early (for example one clock edge too early), or the output signal could become unstable and not usable. This way the hold time limits the maximum input time for the flip-flop.

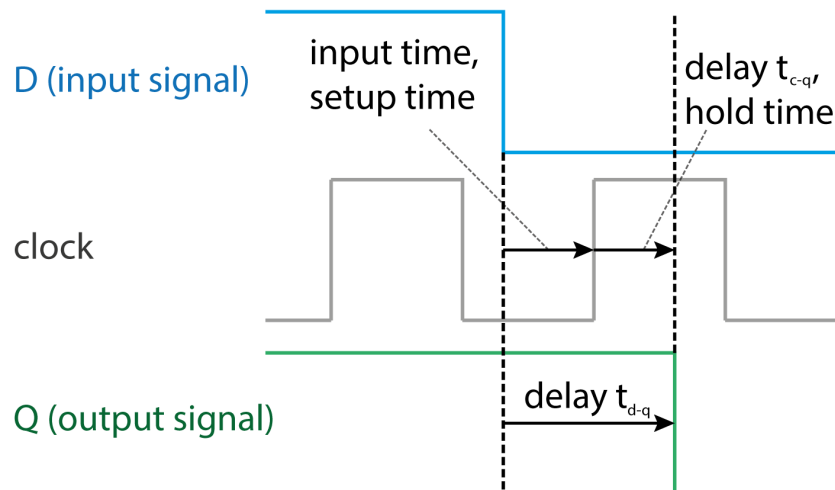


Figure 4.1: An illustration showing some of the key parameters related directly to the signal input and output of a flip-flop.

### Transistor count

The *transistor count* is simply the number of transistors that is needed in order to make the specified flip-flop. It is a good measure for comparison of the area usage of the different flip-flop designs. It is far from an accurate measurement of chip area since many other factors like transistor size and other layout considerations will put constraints on area. However the transistor count gives a good impression of what one can expect for comparison. A small area is obviously better, although it varies a lot whether the area is a design constraint or not.

## Yield

Another parameter that is of interest is the *yield* which is related to *robustness and mismatch*. These two can often be a problem at low voltage operation. If there is a large mismatch the production *yield* will decrease because a larger number of devices will be incapable of functioning normally. Similarly robust circuits will experience less problems from variations and hence will have a higher yield. A high yield is important to ensure that the production will be economically profitable. The yield is a number, usually given in percent, that describes how many of the produced circuits are of high enough quality to be used/sold.

## Energy Delay Product

The *energy delay product*(EDP) is an important factor for delay and energy considerations. The EDP is an indicator of how energy efficient a circuit component is. It is defined as  $PD^2$ , where  $P$  is the power consumption of the component and  $D$  is the delay. There are different ways to measure power and delay, however the specific way of obtaining the EDP for this thesis is discussed in more detail in section 6.

## Differentiality

Another factor to take into consideration is whether or not the flip-flop is *differential*. Differential circuits has inverted outputs and inputs in addition to the normal outputs and inputs. A differential flip-flop will usually be more robust and resistant towards mismatch than a non-differential one, in part because of the symmetry of the layout design. It will also usually be more robust towards external noise.

## 4.2 Standard flip-flop designs

It would be convenient to have a reference flip-flop for comparison. This would make it easier to evaluate the quality of each ULV flip-flop that will be presented in this thesis. The numbers used for comparison are based on simulations done in the same CMOS technology and test environment as the ULV flip-flops. The only difference is that the simulation frequency was reduced to 10MHz for the reference flip-flops because some of them did not function very well on higher frequencies.

### 4.2.1 Conventional Flip-Flop

While there are many flip-flop designs available for comparison there are not many flip-flop designs which focus on ULV and high speed. It turns out that a *conventional transmission-gate flip-flop* (CTGFF) which uses only inverters and transmission gates, could in fact be a suitable comparison for the purpose of this thesis. This is because most modern low voltage flip-

flop designs are targeted towards low power and not high speed which is the focus of this thesis.

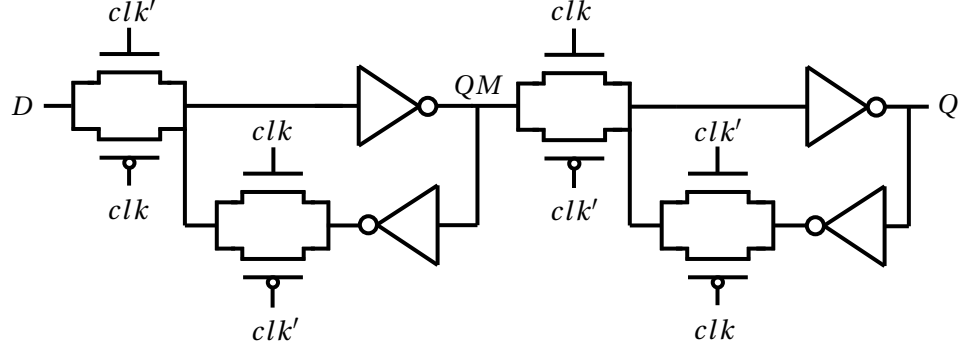


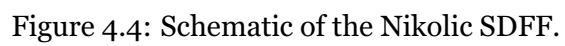
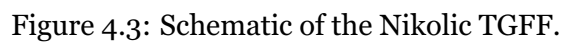
Figure 4.2: Schematic of the CTGFF.

This basic flip-flop design works in a very straight-forward way. The input signal passes through the first transmission gate and is inverted at  $QM$ . When the clock edge arrives and the lock signal is switched the first set of transmission gates are turned off and the second ones open up. The input state right before the clock edge arrived will be stored in the first set of inverters. The  $QM$  signal is now sent into the next two inverters where it is stored, and a replica of the  $D$  signal right before the clock edge is created at the  $Q$  output. The input cannot interfere with this output because the input transmission gate is closed. When the next clock edge arrives and the clock switches back to it's first state, the second set of inverters holds the output signal state.

#### 4.2.2 Nikolic flip-flops

The flip-flop designs investigated by Nikolic [26] are mostly for low-power design. Out of the flip-flop designs compared in the article, the one with the lowest delay seems to be the *semi-dynamic flip-flop* (SDFF)[29]. However for very low supply-voltages the *modified sense amplifier based flip-flop* (MSAFF)[30] was shown to have an even lower delay. The *transmission-gate flip-flop with input gate isolation* (TGFF)[31] is the one that is recommended for low-power use and as such will also be taken into consideration as a possible reference flip-flop. The TGFF very closely resembles the CTGFF while the SDFF works in a different manner than the other two.





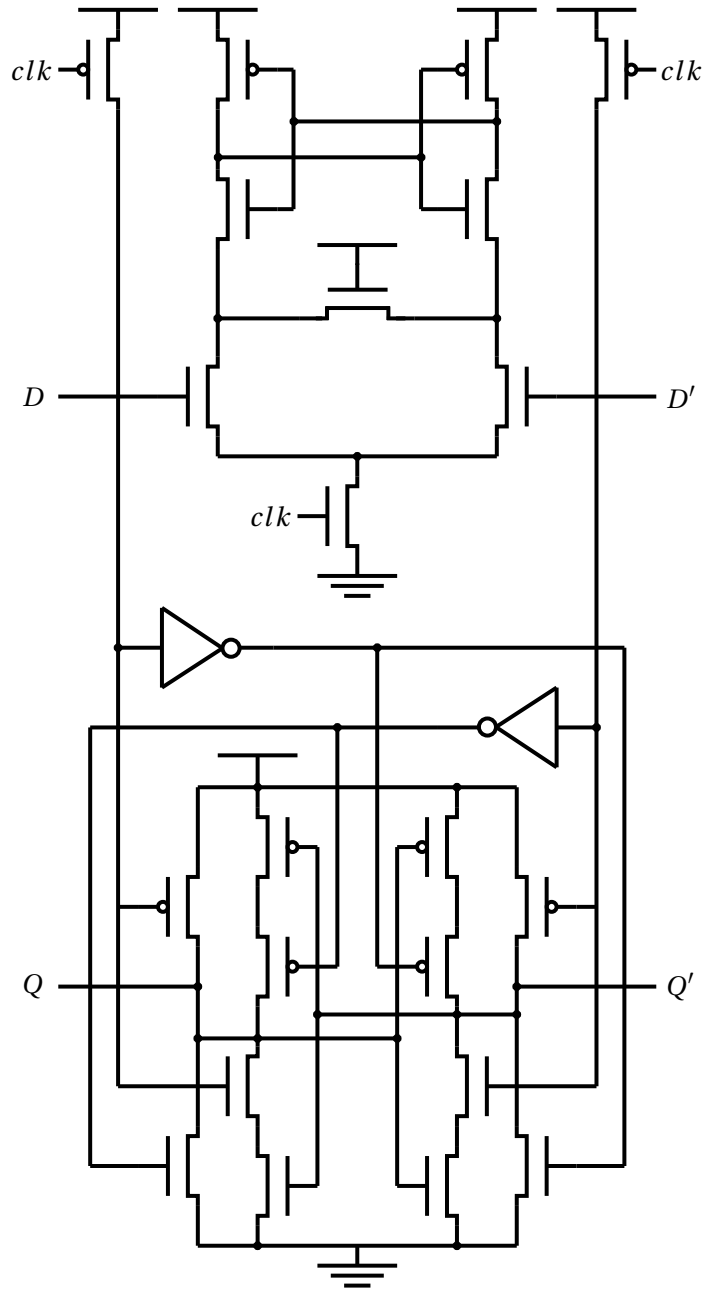


Figure 4.5: Schematic of the Nikolic MSAFF.

### 4.2.3 Comparison

Flip-flop	$t_{setup}$	$t_{hold}$	$t_{c-q}$	$t_{d-q}$	Transistors	Differential
CTGFF	8.3ns	0ns	8.1ns	16.4ns	16	no
TGFF	12.0ns	0ns	19.0ns	31.0ns	20	no
SDFF	0ns	9.0ns	4.5ns	4.5ns	23	no
MSAFF	0ns	2.6ns	10.0ns	10.0ns	26	yes

Table 4.1: Comparison between the different possible reference flip-flops at 300mV.

The simulations to compare the different flip-flops were done and the results can be seen in Table 4.1 where all key parameteres for delay considerations are shown. Two of the flip-flops suffer from setup time limitations while the other two suffers from hold-time limitations.

It can be seen that the input gate isolation used for the TGFF has a high cost in delay-time if it's compared to the CTGFF which is about twice as fast. This is in addition to the cost of 4 extra transistors. It turns out that the flip-flop topology which in Nikolic [26] is considered to be the best general choice for low-power operation is not a good choice if high speed is important. In fact a conventional flip-flop design is better. The SDFF is the best one considering input to output delay which could have made it the best candidate for a high speed flip-flop, however it suffers from a very large hold time which makes it much less desirable. The MSAFF which was found to have a very low delay in Nikolic [26] at very low supply voltages proves itself as a suitable high speed comparison. It has the highest cost in number of transistors used, but the input to output delay is beaten only by the SDFF. Furthermore it has a hold time which is much more acceptable than the SDFF making it the best candidate for a low power high speed flip-flop comparison. As a last advantage this design is the only one that is differential making it suitable as a comparison because differential flip-flops is what dominates this thesis.

The MSAFF will be used as a reference flip-flop for this thesis and will as such be seen in future comparisons.

## 4.3 ULV flip-flops

We will start off with the basic ULV flip-flop shown in Figure 4.6 as originally presented in [10], and then we will look at several variations based on the basic ULV flip-flop. The flip-flops are very similar and use the same principles. We will do schematic simulations on 5 different ULV flip-flops and then choose one for further investigations with layout and chip design. The flip-flops have been labeled alphabetically from flip-flop A to E.

#### 4.3.1 Basic ULV flip-flop (FFA)

We will look at the non-differential and differential basic ULV flip-flop at  $300mV$ , and then compare the results. The schematics of the non-differential and differential flip-flops are shown in Figure 4.6 and 4.7. Where  $E_{n/p}$  are the evaluate transistors and  $R_{n/p}$  are the recharge transistors. These flip-flops are designed to switch at rising clock edges and have inverted outputs ( $Q$  follows  $D'$  and vice versa). Simulations were done with two different frequencies,  $50MHz$  and  $12.5MHz$ .

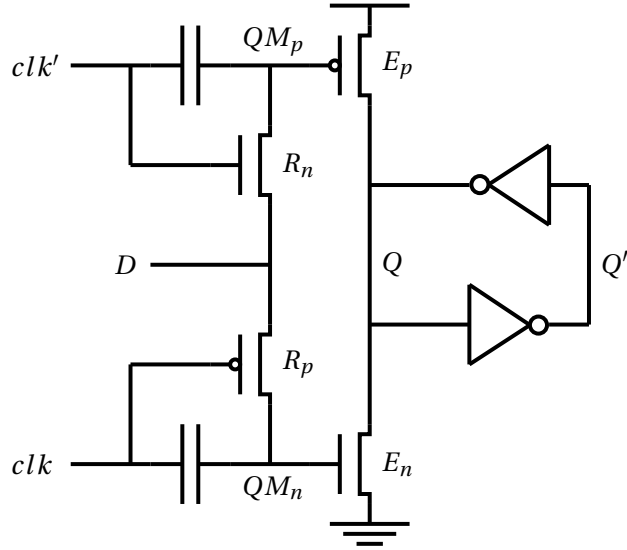


Figure 4.6: Schematic of the non-differential basic flip-flop.

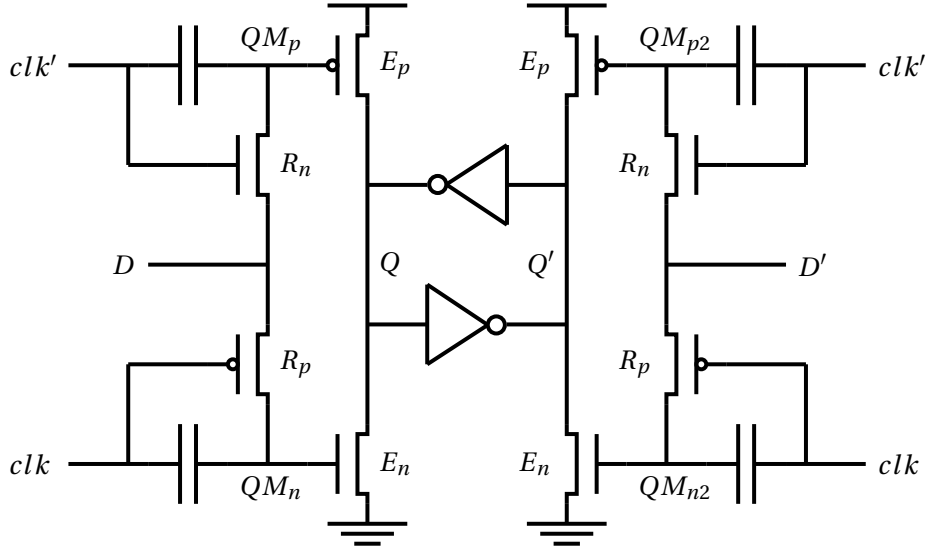


Figure 4.7: Schematic of the differential basic flip-flop.

From Figure 4.8 it can be seen that  $50\text{MHz}$  is too high. The output signal  $Q$  does not respond correctly to switch the inverted output signal  $Q'$ . If we look at Figure 4.9 the frequency is four times lower and it can be seen that it's starting to resemble a flip-flop output and the frequency is not the main issue. The output signal starts to switch before the rising clock edge. We can see, for both frequencies, that when the clock signal is high, the output signal drops. This issue will be dealt with later in this section.

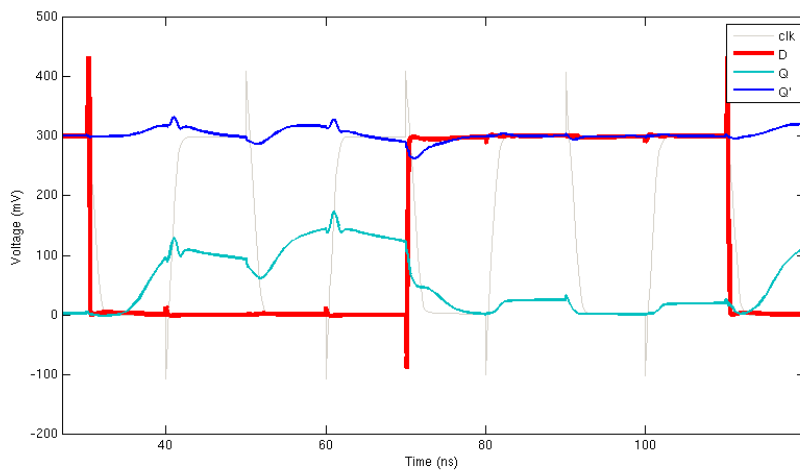


Figure 4.8: Simulation of the basic non-differential flip-flop at  $50\text{MHz}$ .

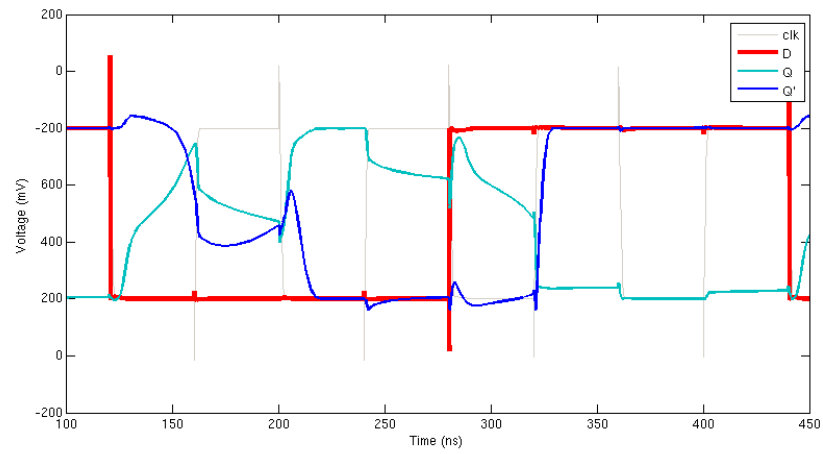


Figure 4.9: Simulation of the basic non-differential flip-flop at 12.5MHz.

If we look at the differential solution, it can be seen from Figure 4.10 and 4.11 that the output signal rise too early for both frequencies, just as they did with the non-differential solution. This is because the  $D$  signal is low and the inverted clock signal is high which gives the pMOS evaluate transistor a low gate signal and hence the output signal goes high. The higher the input time, the earlier it rises relatively.

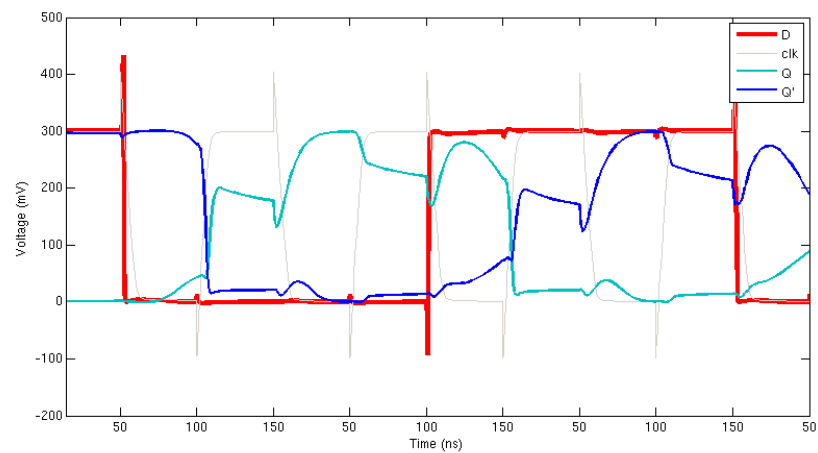


Figure 4.10: Simulation of the basic differential flip-flop at 50MHz.

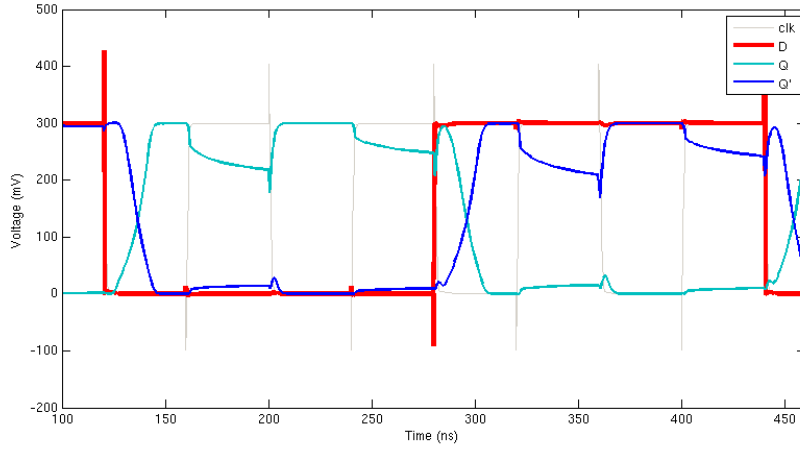


Figure 4.11: Simulation of the basic differential flip-flop at 12.5MHz.

There is a big difference between the differential and non-differential flip-flop results. The non-differential flip-flop is not even close to a satisfying result. A big problem with the non-differential solution is that  $Q$  and  $Q'$  are not completely inverted signals as they are with the differential solution. The differential flip-flop can also work at higher clock frequencies. Because of these results we will from now on only look at differential solutions.

As mentioned earlier there was a problem with the output signal when the clock signal is high. To look closer at this issue we have to look at the  $QM_n$  and  $QM_p$  signals which are the inputs for the evaluate transistors. Figure 4.12 shows a close up of one clock pulse at a frequency of 3.125MHz. In this clock pulse the input signal  $D$  is low and the output signal  $Q$  should be constantly high which is not the case. The reason for this is because when the clock goes high it drives  $QM_n$  above  $V_{dd}$  which gives the nMOS evaluate transistor a boosted voltage (ca. 350mV). At the same time the pMOS evaluate transistor gets an even bigger boosted voltage (ca. -150mV), but it's not enough to negate the effect from the  $QM_n$  signal. We will present a solution for this issue in later sections.

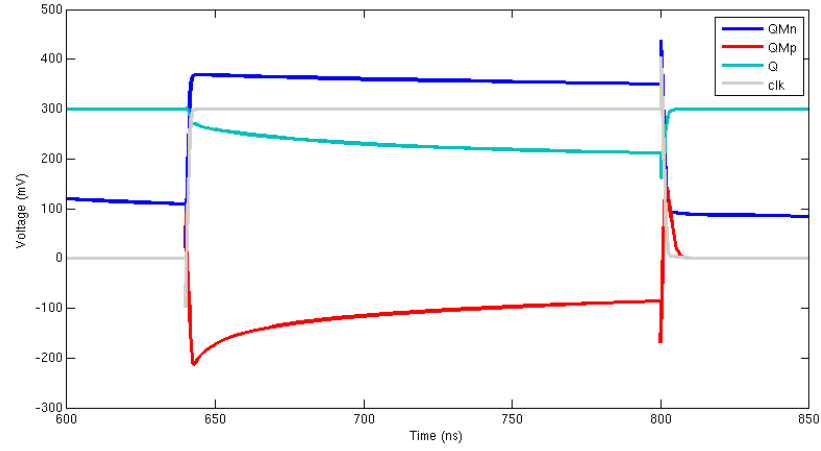


Figure 4.12: Simulation of the basic differential flip-flop at 3.125MHz.

#### 4.3.2 Input considerations for simulations

The input signal  $D$  and  $D'$  will limit the minimum setup time that can be achieved in the flip-flop. This is because this signal is responsible for recharging the  $QM$  nodes which creates the boost that allows the flip-flop to operate properly. So if this signal is very strong it can charge up the node faster effectively reducing the minimum setup time. If this is the case then the recharge transistor  $R_{p/n}$  will still limit the amount of charge that the  $D$  signal can deliver to the  $QM$  node. This transistor in relation to setup time will be discussed in more detail later. Realistically the  $D$  signal will not be very strong and will set a limit to the minimum setup time that can be achieved, which results in a maximum effective transistor strength for  $R_{p/n}$  which is entirely dependent on the strength of the input signal.



## The recharge transistor

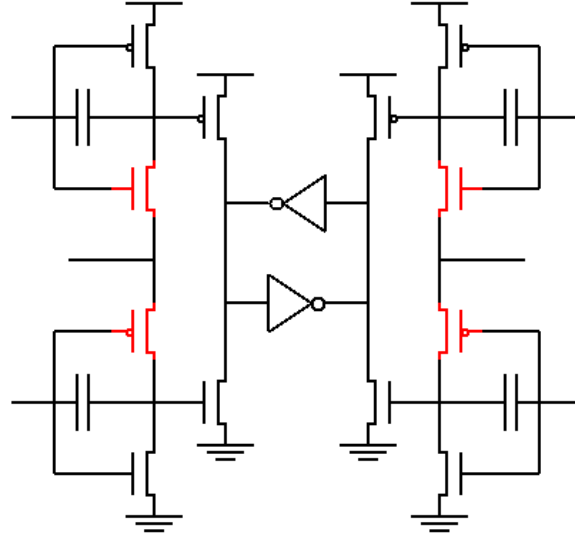


Figure 4.13: Generic ULV flip-flop with recharge transistors marked in red.

This transistor works as a switch that lets the recharge signal onto the  $QM$  node. For the ULV flip-flops the recharge signal will be the input signal  $D$  or  $D'$ . The recharge transistors are generally located as shown in Figure 4.13.

As previously explained the recharge transistor  $R_{p/n}$  plays a role in determining the minimum setup time that can be acquired. This is because this transistor delivers the charge to the  $QM$  node used to charge it up, so the more charge  $R_{p/n}$  delivers the faster the node will be charged up reducing the setup time.

Normalized transistor strength $R_{p/n}/R_{p/n,min}$	Normalized setup time
1.0	1.00
2.0	0.75
3.0	0.58
5.4	0.47
8.0	0.25

Table 4.2: The effect of the recharge transistor strength on flip-flop setup time at  $300mV$ , where a strength of 1 represents the minimum transistor strength.

The transistor strength values in Table 4.2 are approximate values that shows that the more current it is capable of delivering the less setup time is required. The strength levels were achieved using the previously discussed techniques for tuning transistor strength at low voltage as seen in Table 3.1. The techniques used for tuning the transistor strength are mainly body-biasing and transistor sizing as discussed earlier. Transistor sizing in

particular will have an effect on the capacitive connection at the  $QM$  node, which in turn will have an effect on the setup time. However the change will be relatively small compared to the change in setup time as a result of recharge transistor strength. This does however mean that the values given in Table 4.2 can fluctuate a little. As mentioned previously the input signal  $D$  will also set an additional limit to the setup time.

The recharge transistor will also affect the hold time. The limitations of hold time will be discussed in more detail in the next section. If the recharge transistor is weak this limitation will be much less of a problem. However we want to reduce the setup time as much as possible, therefore a large recharge transistor is used throughout this thesis with a transistor strength of 8.

### 4.3.3 Flip-Flop inverters

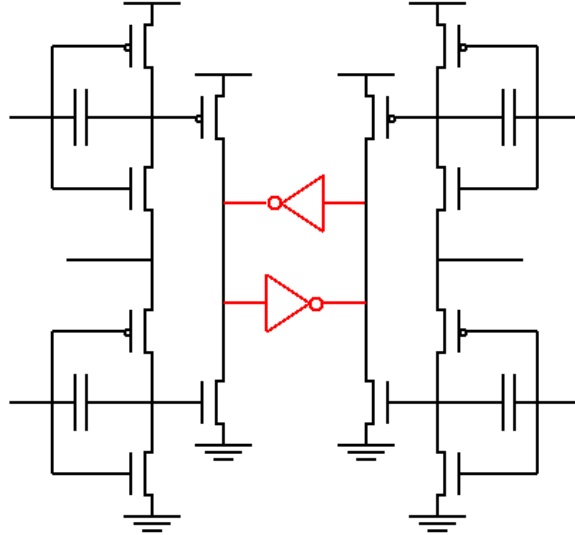


Figure 4.14: Generic ULV flip-flop with the flip-flop inverters marked in red.

In the flip-flops there are 2 inverters (shown in Figure 4.14) responsible for storing the output signal  $Q$  and inverted output signal  $Q'$ . In this flip-flop design these inverters are not responsible for switching, and will only be used for storing. Therefore these inverters do not need to have a fast switching time. The switch is done by the evaluate transistors that are boosted with  $\frac{3}{2}V_{dd}$  for nMOS and  $-\frac{1}{2}V_{dd}$  for pMOS so that these transistors will have much more strength during switching than the storage inverters.

It has to be ensured that the stored value will not be switched before the correct clock edge event, for example by the  $D$  signal going high at the  $QM$  node of an nMOS evaluate transistor during the recharge phase. This is accomplished by making the inverters strong enough to override an evaluate gate that has not received a boosted gate input. At the same time the inverters has to be weaker than an evaluate transistor experiencing a

boosted gate input.

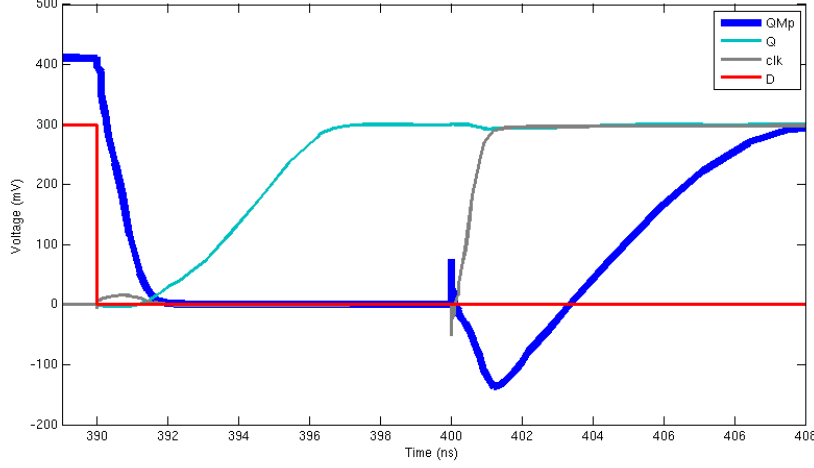


Figure 4.15: Simulation at  $300mV$ , showing what happens if the inverter strength is too low. Note that this flip-flop has been stabilized (using a keeper transistor which will be explained later).

The effects of having flip-flop inverters that are not strong enough is shown in Figure 4.15. Once the  $D$  signal goes low it causes the  $QM$  node voltage to go low. The pMOS evaluate transistor is turned on, and slowly switches the output causing the output signal to follow the input signal instead of waiting for the clock edge. The flip-flop becomes transparent and does no longer function properly. This problem occurs for longer input times, and not when the input time is small. At small input times the output signal will not switch too early because it simply does not have enough time.

The flip-flop will at lower inverter strength levels experience a maximum input time. This is a maximum amount of time where the input signal can switch while the output still remains unchanged until the right clock edge arrives in the evaluate phase. This can be viewed as a hold time that is dependent on the frequency. The hold time can be expressed as:

$$t_{hold} = \frac{1}{f} - t_{max}$$

Where  $f$  is the clock frequency used and  $t_{max}$  is the maximum input time. When  $t_{max}$  is much smaller than  $\frac{1}{f}$ , there will be a long hold time limit for the flip flop. When  $t_{max} > \frac{1}{f}$  the hold time will be gone. Because of this the hold time is only an issue for low frequencies. Therefore there is a trade-off between setup time and hold time. If the inverters are very strong the input time can also be longer since the evaluate transistors will not override the inverters easily. However this will also make it harder for a boosted evaluate transistor to switch the stored value since the inverters are more difficult to override. This means that the gate capacitor has to be

charged longer to get a strong enough boost to switch the signal, thereby creating a longer setup time. A bigger problem that occurs is that this will also increase the switching time since the evaluate transistors have to drive a bigger load.

The optimal inverter to evaluate relation will be investigated in the next section.

#### 4.3.4 The evaluate transistor

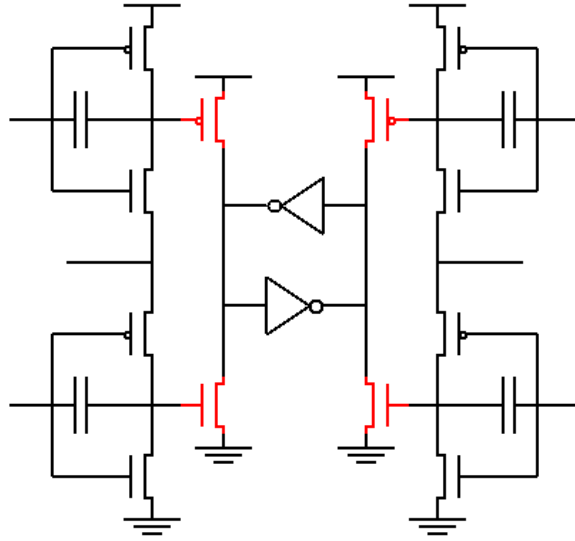


Figure 4.16: Generic ULV flip-flop with the evaluate transistors marked in red.

The evaluate transistor is responsible for setting the output value of the flip-flop. The strength of this transistor has to be set according to the previously discussed implications of the inverters. While it might be desirable to have these transistors as strong as possible in order to decrease the overall switching time of the circuit, this will also increase the strength requirement of the inverters. This has to be taken into consideration so that inverter strength and evaluate transistor strength are in a good balance.

Normalized inverter strength inv/eva	2	3	4	6	9
Normalized $t_{c-q}/t_{min}$	1.0	1.0	1.0	2.0	4.6
Normalized $t_{setup}/t_{min}$	2.8	2.8	2.8	4.0	20.0
Normalized maximum input	8.0	30.0	$\infty$	$\infty$	$\infty$

Table 4.3: Normalized clock-to-output delay and input limitation for different normalized inverter strengths at 300mV (the evaluate transistor strengths apply for hold mode, not during a boosted switch).

In Table 4.3 the effect of proper balancing between the inverter and evaluate transistor strength is shown. The data collected assumes a strong

recharge transistor ( $7.5 \times \text{evaluate}$ ) since this is where the maximum input limitation is most problematic. At low recharge transistor strength levels the optimal inverter size is lower.

It can be seen from Table 4.3 that the inverter strength needs to be higher than the evaluate transistor strength, so that the hold time will be reduced. If the inverter strength is high enough compared to the evaluate strength, the evaluate transistor will be incapable of switching the output without a boosted gate input. Hence the maximum input limit is removed (seen in Table 4.3 as  $\infty$  maximum input).

Furthermore it can be seen that when the inverter strength becomes too large, the flip-flop starts to experience decay in the switching time as well as in setup time.

From the simulations done in this project it would appear that the inverters should be about four times as strong as the evaluate transistors. This way there is virtually no decay in the switching time  $t_{c-q}$  nor in the setup time, while at the same time the limitations on maximum input is removed.

#### 4.3.5 Flip-flop B (FFB)

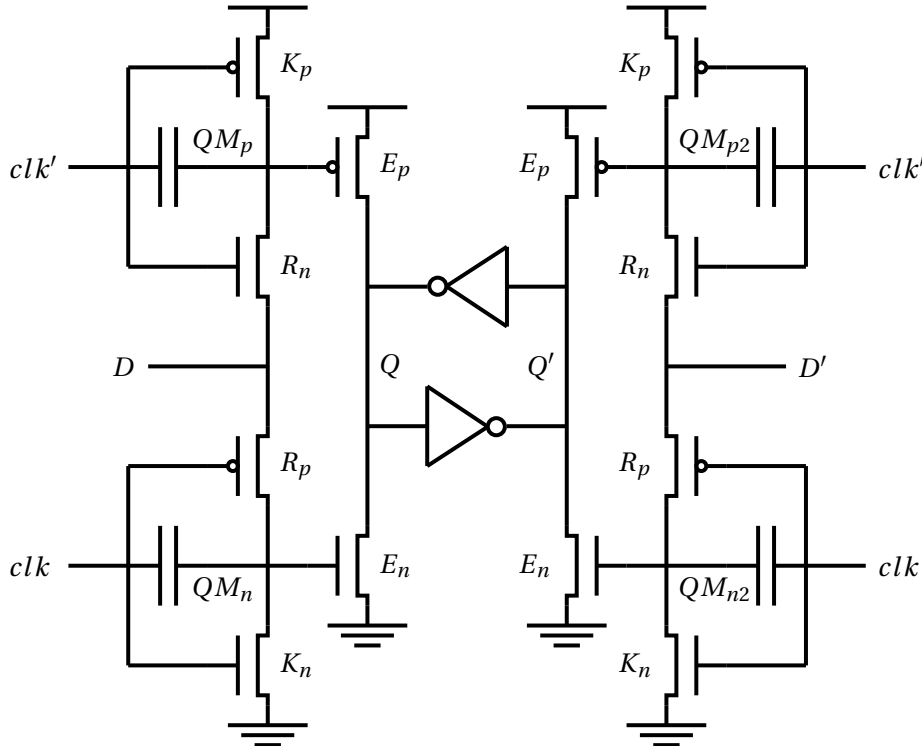


Figure 4.17: Schematic of flip-flop B.

The problems that were present in the basic ULV flip-flop will be solved in this section. The problem occurs because there is enough leftover charge at the QM node from a previous switch/transition to partially or completely

drive a switch at the wrong evaluate transistor. This can cause the output signal to become weaker or unstable, or in the worst case even switch to the wrong value.

In this variation of the ULV flip-flop and all following variations a keeper transistor ( $K_p$  or  $K_n$ ) has been added to the  $QM$  nodes as seen on Figure 4.17. This keeper transistor serves as a tool for emptying the charge after a switch so that nothing is left over to distort the input from the next clock period.

Additionally this keeper transistor serves as a tool for turning off all evaluate transistors during the evaluate phase (but not on the transition from recharge to evaluate). In this way there is no contest between the evaluate transistors and the inverters, and as a result the output signal is much more stable and stronger.

During the recharge phase the keeper is turned off. This is achieved by connecting the gate to the appropriate clock signal. Since the keeper empties the charge at the  $QM$  node preventing the unintentional switching of the evaluate transistors, the keeper only needs to be turned off if the input signal  $D$  or  $D'$  is in a state where it will charge up the  $QM$  node. That is to say that the keeper needs to be turned off so that charge can actually build up on the  $QM$  node during the recharge phase. This means that it is also possible to connect the keeper gate signal to the input signal turning it off only when the input requires it. However this would cause a delay in turning the keeper off which could increase the setup time. In addition this requires the input signal to drive an additional gate node, which in turn increases the input load of the flip-flop.

At the transition from the recharge phase to the evaluate phase the evaluate transistors will give a short burst of increased  $I_{on}$  current. An important consideration to keep in mind when using the keeper transistors is the transistor strength. The stronger the keeper is the less powerful the  $I_{on}$  current of the evaluate transistor will be. This effect is minimal as long as the keeper is fairly weak (minimum size) because the keeper is turned on much slower than the evaluate transistor thanks to the boosted voltage level at  $QM$ . This way the charge at the  $QM$  node is not emptied before the evaluate transistor has a chance to switch the output.

During the evaluate phase the keepers keeps the evaluate transistors in off mode, and ensures that the  $QM$  nodes are charged with the opposite value of what is needed to turn on the evaluate transistors.

Finally on the transition from evaluate to recharge the keepers are turned off in preparation for the recharge phase.

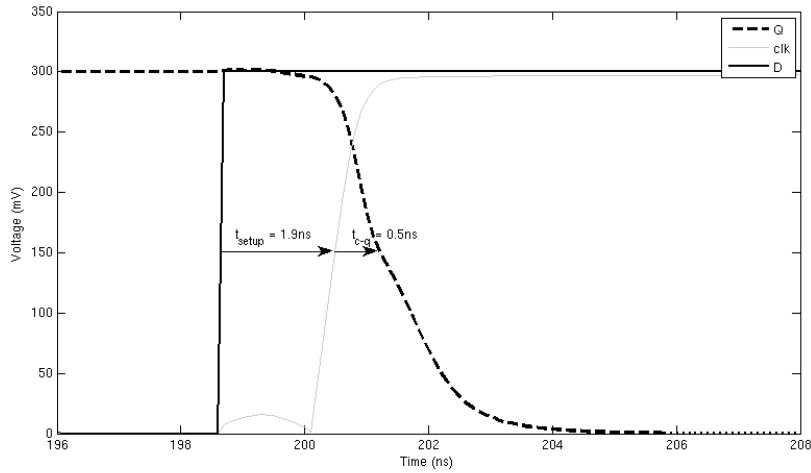


Figure 4.18: Simulation of a transition of flip-flop B at 300mV.

Flip-flop	$t_{setup}$	$t_{hold}$	$t_{c-q}$	$t_{d-q}$	Transistors	Differential
MSAFF	0ns	2.6ns	10.0ns	10.0ns	26	yes
FFB	1.9ns	0.0ns	0.5ns	2.4ns	16	yes
Adds $t_{setup}$ No $t_{hold}$ 20.0× 4.2× -38% area						

Table 4.4: Improvement compared to the reference flip-flop at 300mV.

Simulations were done comparing this flip-flop with the reference flip-flop as seen in Table 4.4. These simulations were done using strong recharge transistors and an inverter strength of  $4 \times evaluate$ . It was shown earlier that this gives good setup and delay conditions for ULV flip-flops. These same conditions will also be used on following flip-flop designs.

From Table 4.4 it is clear that FFB is much more superior to the reference flip-flop in almost all aspects. The only disadvantage it has is the added setup time, however this is easily made up for by the 20 times faster switching time. In total the input-to-output delay is reduced enough to make FFB more than four times faster than the reference flip-flop.

In addition to the speed increase offered by this flip-flop design and all following ULV designs, there is also an area gain. This area gain can for example be used to increase transistor strength by sizing everything up, resulting in a reduction in switching time. Alternatively the extra area can be used to reduce internal noise by spacing the transistors further apart. Keep in mind that there will be additional capacitors present in the ULV flip-flop layout design. This adds some area, although these capacitors can be implemented entirely into an unused metal layer as will be seen in chapter 8.

#### 4.3.6 Flip-flop C (FFC)

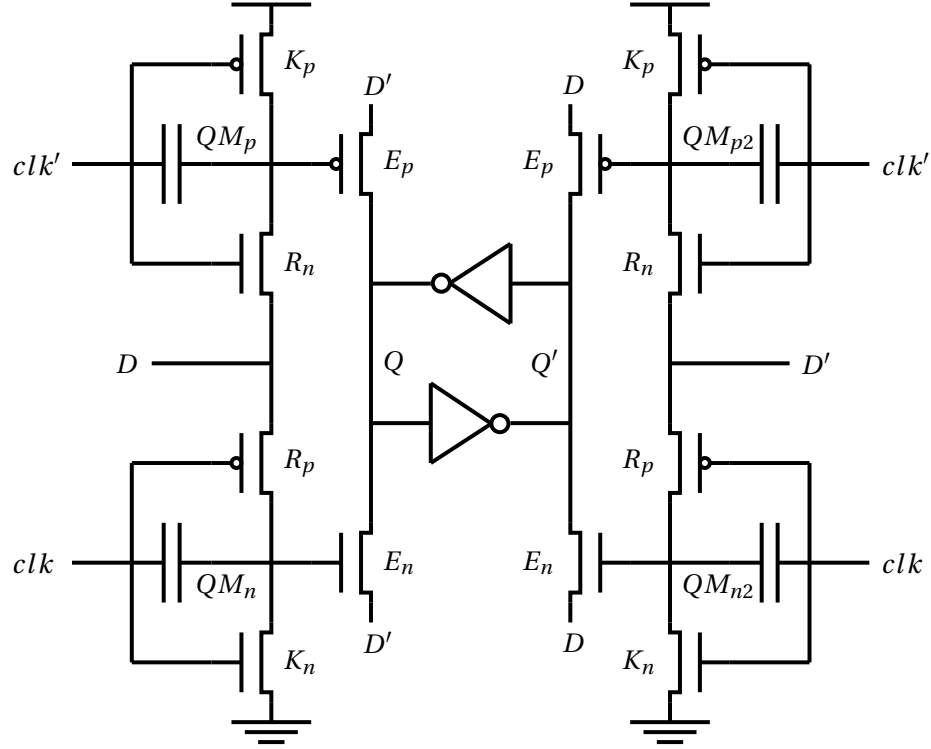


Figure 4.19: Schematic of flip-flop C.

It would seem more appropriate to apply the  $D$  signal directly into the  $Q$  signal at rising clock edge when latching happens. This way the output will latch directly onto the input. This means that both the nMOS and pMOS evaluate transistors are transmitting the same signal into the output node. This way there is no contest between the evaluate transistors pull-down and pull-up. This should reduce the amount of boost that is needed in order to get a proper switch on the output, which in turn will reduce the setup time.

However if this design is to be used it has to be ensured that the input signals  $D$  and  $D'$  are sufficiently powerful to drive both the four QM nodes and the four evaluate transistor sources.

Flip-flop	$t_{setup}$	$t_{hold}$	$t_{c-q}$	$t_{d-q}$	Transistors	Differential
MSAFF	0ns	2.6ns	10.0ns	10.0ns	26	yes
FFC	1.0ns	4.0ns	0.7ns	1.7ns	16	yes
Adds $t_{setup}$		0.7×	14.3×	5.9×	-38% area	

Table 4.5: Improvement compared to the reference flip-flop at 300mV.

In Table 4.5 we can see that indeed the setup time is lower than for FFB. It is now only 1.0ns where as FFB has a setup time of almost twice as much. However it comes at a small cost on the clock-to-output delay time.



This is a result of the change in the source nodes of the evaluate transistors which now is variable, and also slightly limited in strength by the input signal. In FFB these nodes receive the very strong signals from power and ground which allows for much more current to flow hence making the switch stronger.

In total this flip-flop switches 14 times as fast as the reference flip-flop, but more importantly has an input to output delay that is 6 times faster. FFB was only about 5 times as fast as the reference flip-flop, so there is clearly an overall improvement in speed.

A significant loss from this design is the added hold time that did not appear in the previous design. The input signal needs to remain stable while the evaluate transistors are still boosted since they are now directly connected to the output. If the input signals change before the boost is over the evaluate transistor will be able to drive this change onto the output. The added hold time is quite significant and even longer than that of the reference flip-flop.

The added hold time and the big requirement on input signal strength makes this design an unlikely choice for use in a bigger circuit. However if the input signal is known to be of significant strength and hold time issues are not that important, then this design certainly is a good choice.

### 4.3.7 Flip-flop D (FFD)

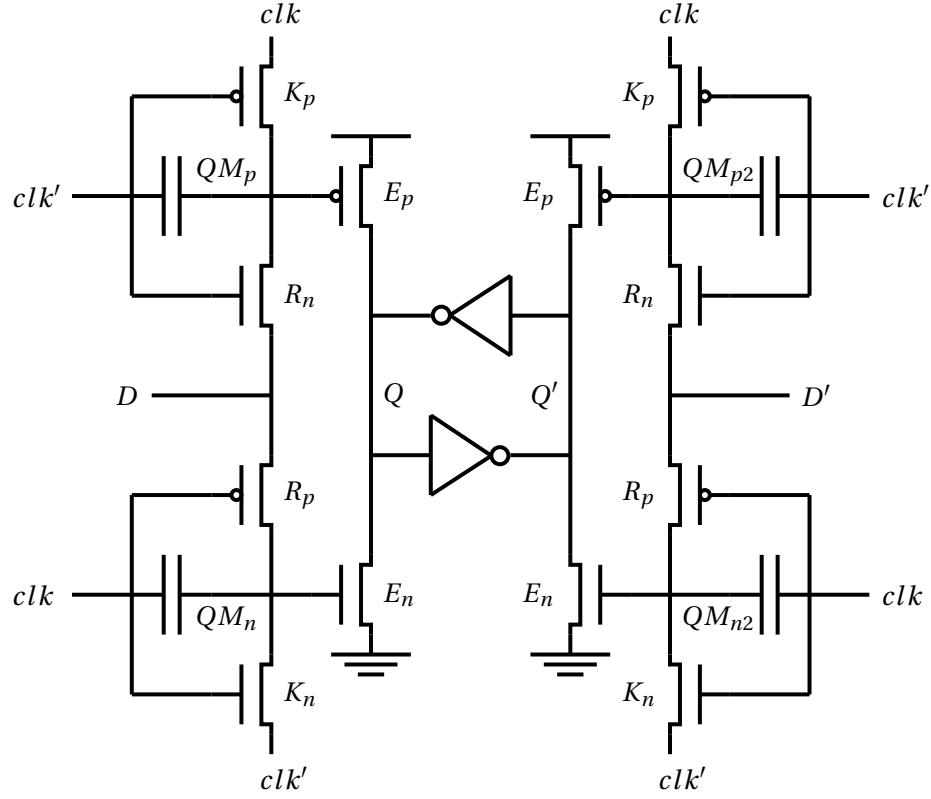


Figure 4.20: Schematic of flip-flop D.

The only difference between this variation of the flip-flop and FFB is the source for the keeper transistors. Instead of using the supply voltage the clock has been used. This has been done to show that this is just as effective as using  $gnd$  and  $v_{dd}$ . From Table 4.6 it can be seen that the results are almost identical to FFB with a minimal increase in setup time. Since the gate and the source signals are always inverted of each other, on the keeper transistor, it will work in the same way as using  $gnd$  and  $v_{dd}$ . On the next flip-flop we are going to present we will see that the gate and source signals are not necessarily inverted of each other.

Flip-flop	$t_{setup}$	$t_{hold}$	$t_{c-q}$	$t_{d-q}$	Transistors	Differential
MSAFF	0ns	2.6ns	10.0ns	10.0ns	26	yes
FFD	1.8ns	0ns	0.5ns	2.3ns	16	yes
Adds $t_{setup}$ No $t_{hold}$ 20.0× 4.3× -38% area						

Table 4.6: Improvement compared to the reference flip-flop at 300mV.

It can be seen from Table 4.6 that There is virtually no difference between FFD and FFB so they can both be used if desered depending on

other factors than speed.

#### 4.3.8 Flip-flop E (FFE)

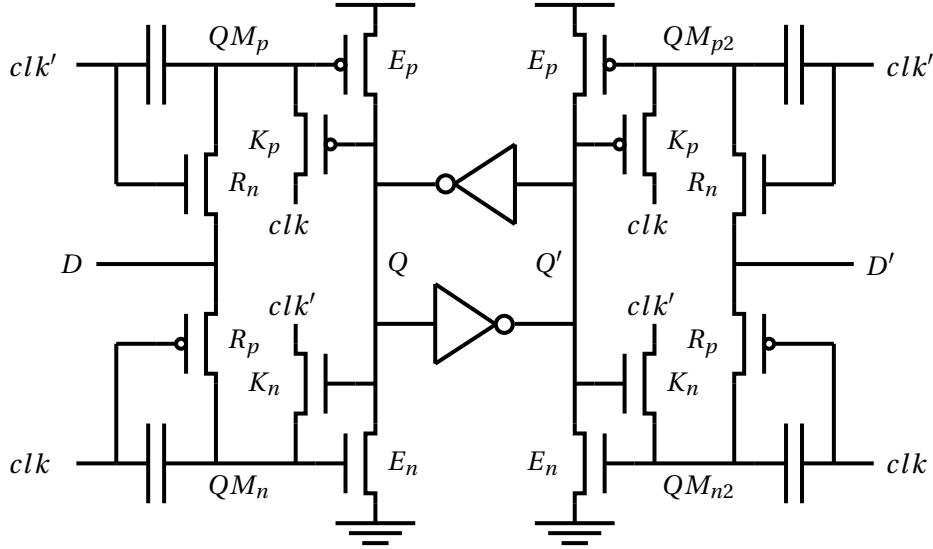


Figure 4.21: Schematic of flip-flop E.

A feedback loop can be used to control the keeper transistors. Looking at the left side of the differential flip-flop this feedback loop works as follows:

##### 1. $Q=1$ , $D$ transitions from 0 to 1

During the evaluate phase the keepers should be preventing the output from switching. If  $Q$  is high only the nMOS keeper will be on. This keeper transmits the low signal from the inverted clock into the  $QM$  node turning the nMOS evaluate transistors off. This prevents the nMOS evaluate transistor from leaking a low signal from ground into the output signal  $Q$  which is supposed to stay high. The pMOS keeper does not help turn off the pMOS evaluate transistor like it did in previous flip-flop designs, however this is not a problem since this evaluate transistor delivers a high signal from the power supply which does not interfere with the output signal which is supposed to stay high. In fact not turning on this keeper could potentially help the output signal to be more stable during the evaluate phase. The downside is that the keepers need to wait for the output signal instead of switching at the clock flank. This adds a small hold time to the flip-flop.

Once the recharge phase starts the pMOS keepers will still be turned off while the nMOS keepers are turned on. The only change on the keepers then is the change of the clock signal going through the

keepers, since the clock signal has switched. Again this does not affect the  $QM_p$  node since the keeper is turned off at this node, but at the  $QM_n$  node the keeper now transmits in a high signal from the inverted clock. This actually helps the transistion for this case, since the input signal  $D$  is going high anyway. Since the  $QM_p$  node was never emptied of charge by a keeper there is a possibility that there will be residue low charge left at this node from when the input signal  $D$  was low in the previous recharge phase. This problem is a small one since the now switching  $D$  signal will try to negate this charge preventing a boosted current from the pMOS evaluate transistor on the positive clock flank.

## 2. $Q=1, D=0$

In this case the output signal should just stay high since there is no change in the input. During the evaluate phase the case is the same as what was discussed in the previous section. However during the recharge phase there is now a possible source for problems where the nMOS keeper is transmitting a high signal to the  $QM_n$  node from the inverted clock, while the input signal  $D$  is trying to pull down the value at this node. Because of this it is important that the keeper transistors are not too strong compared to the recharge transistors. As will be seen later this also reduces the robustness of this flip-flop as a result of mismatch and other variations.

## 3. $Q=0, D$ transitions from 1 to 0

This case is just the opposite of case 1 and works in the same manner.

## 4. $Q=0, D=1$

This case is just the opposite of case 2 and works in the same manner.

For the right side of the differential flip-flop the functionality is the same except inverted.

This kind of feedback does add a little bit more load to the output which is something that should be taken into consideration when using this design.

Flip-flop	$t_{setup}$	$t_{hold}$	$t_{c-q}$	$t_{d-q}$	Transistors	Differential
MSAFF	0ns	2.6ns	10.0ns	10.0ns	26	yes
FFE	1.2ns	0.1ns	0.6ns	1.8ns	16	yes
Adds $t_{setup}$		26×	16.7×	5.6×	-38% area	

Table 4.7: Improvement compared to the reference flip-flop at 300mV.

In Table 4.7 the simulation results of FFE can be seen. It is clear that this flip-flop certainly has some good properties for speed considerations. It's about as fast as FFC, however it only suffers from a minor hold time restriction.

While this flip-flop seems to be the overall fastest flip-flop we will later see that the other three topologies are much more robust.

### 4.3.9 Summary table

Flip-flop	$t_{setup}$	$t_{hold}$	$t_{c-q}$	$t_{d-q}$	Transistors	Differential
MSAFF	0ns	2.6ns	10.0ns	10.0ns	26	yes
FFB	1.9ns	0.0ns	0.5ns	2.4ns	16	yes
FFC	1.0ns	4.0ns	0.7ns	1.7ns	16	yes
FFD	1.8ns	0ns	0.5ns	2.3ns	16	yes
FFE	1.2ns	0.1ns	0.6ns	1.8ns	16	yes

Table 4.8: Summary table at 300mV, showing the data for all ULV flip-flops as well as the data for the MSAFF.

A summary of the simulation results for the ULV flip-flops thus far can be seen in Table 4.8. A comprehensive comparison between the different ULV flip-flops can be found in Chapter 7, after yield and power simulations have been presented.

## 4.4 Flip-flops cascaded in shift register

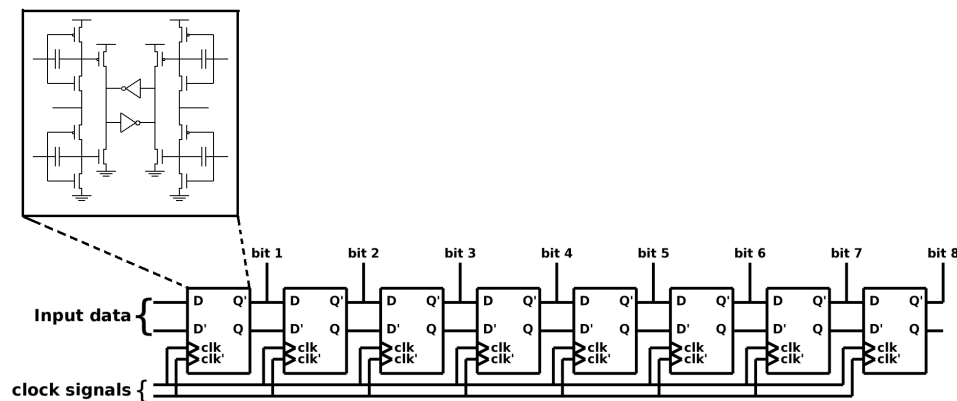


Figure 4.22: An 8-bit shift register constructed from FFB. Registers constructed with other flip-flops will be almost identical with changes only to the outputs and inputs to compensate for whether or not the flip-flop is complementary and how many clock signals it needs.

The ULV flip-flops can be used to implement a shift register by cascading several in a row as shown in Figure 4.22. This shows an example of how the flip-flops can be used and also some of the considerations that are important for such an implementation.

A shift-register is used for storing a stream of bits, which in this case is a single Byte (8 bit). The information can then be accessed from the flip-flop outputs. In addition to the elements shown in Figure 4.22 the clock inputs will be buffered. One would normally add some more logic for activating read/write mode on the register as well.

### Clock leakage into input

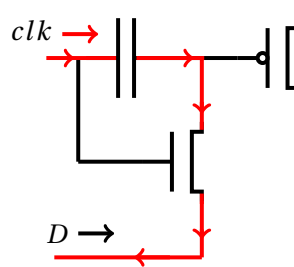


Figure 4.23: The clock leakage into the input signal  $D$ . In a register this  $D$  input signal is the  $Q$  output signal of the previous flip-flop in the register.

Cascading the flip-flops reveals a new problem related to the recharge transistors and input clock. It turns out that the clock signal will go back through the capacitor and the recharge transistor into the input signal  $D$ , creating spikes or instability. In the shift register this means that the output signal from each flip-flop will be unstable, and possibly useless unless countermeasurements are taken.

In order to counter this issue several actions can be taken. Firstly the transistor strength of the recharge transistors can be reconsidered. The weaker this transistor is, the less of the clock signal can travel back into  $D$ . However as described earlier the size of this transistor is crucial for reduction of the setup time of the flip-flops, so a balance needs to be found. Secondly the clock strength can be reduced so that it will have less of an impact. However this will also reduce the  $c - q$  delay, so again a balance needs to be found where the delay is within reasonable bounds, while the clock does not interfere too much with the input signal. Thirdly the strength of the input signal can be increased. For the register this is hard to do since the flip-flop inverters are already as large as they can be without causing a big loss of setup time and delay.

### Input load and output drive

While in a shift register the issue of input load and output drive becomes more specific. One flip-flop needs to be capable of driving the next flip-flop. This means that reducing the input load and increasing the output drive is important in order to achieve a working register.

One way of reducing input load is by reducing the recharge transistor strength. As has already been discussed, this might already be required anyway in order to reduce the leakage from the clock into the input signal. There needs to be a balance of the recharge transistor strength, since if this transistor is too weak, the setup time will be very long. However it needs to be taken into consideration that within a register, the input time requirement can in fact be somewhat less strict. This is because the input signal to a flip-flop will be set as soon as the previous flip-flop has had time to switch. This means that the input time can be as high as

$t_{input} = t_{period}/2 - t_{c-q}$ . This puts less restriction on how weak the recharge transistor can be, allowing it to be reduced more in order to decrease input load.

The output drive can be increased by increasing the flip-flop inverter strength, however these inverters are already about as strong as they can afford to be.

Each flip-flop also needs to be able to drive an additional output circuitry that makes use of the register.

In addition, the first flip-flop in the register needs to have its input load considerations be made according to whatever input it is getting, since it will probably not be driven by yet another ULV flip-flop. The recharge transistor strength and other variables needs to be separately adjusted for this flip-flop.





## Chapter 5

# Robustness, mismatch and variations

In this section we will look at how well all of the different flip-flops perform with process and mismatch variations[23]. With sub-threshold voltage the process variations increase dramatically[32], and therefore it is an important thing to look at when working in the sub-threshold region. Monte Carlo simulations were run in Cadence for each flip-flop with 100 runs per Monte Carlo simulation. The yield will be presented with 5 decades of frequencies ranging from  $10kHz$  to  $100MHz$  at  $300mV$ . Yield will also be presented with different supply voltages, ranging from  $100mV$  to  $400mV$ , to give a thorough analysis of the yield. We will also see how the input time affects the yield.

To find the yield we give the signal a certain time to switch without looking at the actual signal while it switches. Then we check the stability of the signal to see if it stays low or high with a  $\pm 20\%$  margin. We look at three segments as shown in Figure 5.1, after that the signal repeats itself. This allows for easy implementation in MATLAB, so that the analysis process can be automatized for hundreds of simulations.

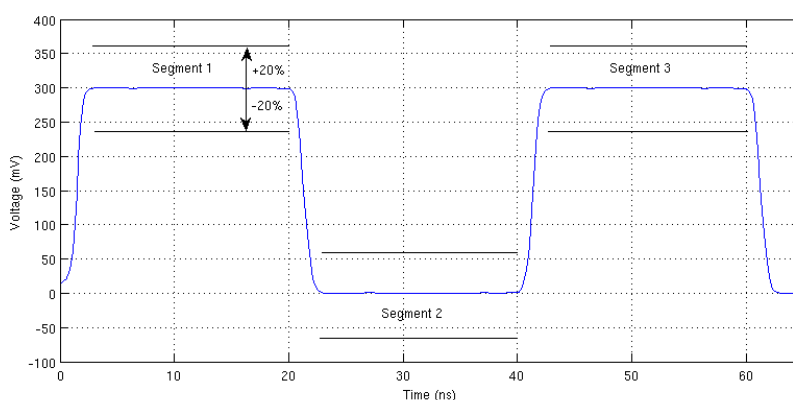


Figure 5.1: Plot illustrating how the yield was found.

## 5.1 Frequency yield

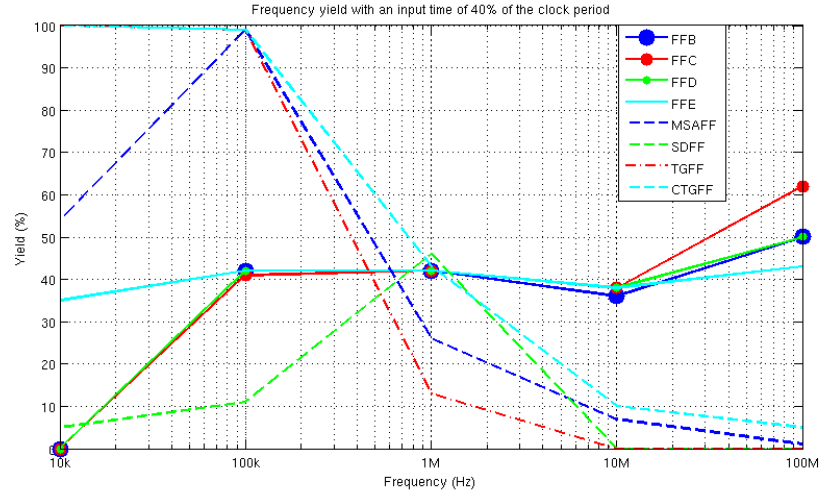


Figure 5.2: Plot showing the yield for all the flip-flops from 10kHz to 100MHz at 300mV.

Figure 5.2 shows the yield for all the flip-flops where the frequency has been swept from 10kHz to 100MHz. The input time used was 40% of the clock period for each frequency. This gives a rather large input time relative to the frequency. Since the input time is an important factor for the flip-flops we want to use a large and a small input time while looking at the yield. As one can see, from Figure 5.2, the ULV flip-flops behave very similarly, but FFE is the only flip-flop which gives any yield at all at 10kHz. FFE is the least frequency dependent flip-flop with a large input time. At 10kHz the clock period is very large thus resulting in a very large input time. With such a large input time the flip-flops don't manage to switch, but by connecting the keepers to the Q signal instead of the clock signal it fixes this issue. When we look at CTGFF and TGFF we can see that they work perfectly well at low frequencies, but the yield drops significantly at frequencies in the mega hertz region and falls below 10% at 10MHz. The long input time doesn't affect these flip-flops at all. The MSAFF works really well at 100kHz, but at 10kHz the input time is too large which gives less yield. It falls apart at higher frequencies just as CTGFF and TGFF. SDFF gives the least satisfying yield at both low and high frequencies except at 1MHz where it gives the highest yield out of all the flip-flops. The ULV flip-flops are the only flip-flops which gives decent yield at 100MHz. They all benefit from the high frequency because of the short input time.

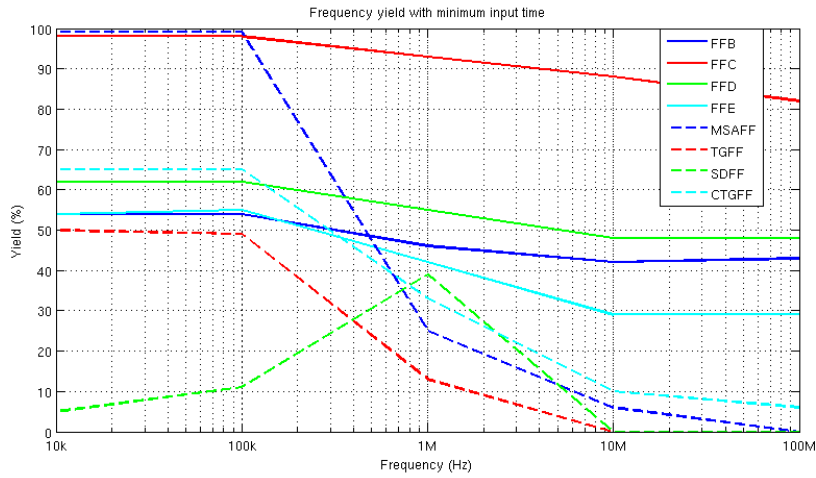


Figure 5.3: Plot showing the yield for all the flip-flops from 10kHz to 100MHz with minimum input time.

By looking at the yield with the minimum input time (the setup time), as shown in Figure 5.3, we can see that all the ULV flip-flops give a better yield at low frequencies, especially FFB, FFC and FFD. The lower frequencies give a better yield than the higher frequencies for all the flip-flops now that the input time is short. FFC also gives a good yield overall and it really benefits the short input time for all frequencies. All the ULV flip-flops, except FFC, gets a worse yield with the setup time as the input time at 100MHz. It's clear that the input time is an important factor for the yield and it affects the flip-flops differently. FFE is more frequency dependent now that the input time is very short while the other ULV flip-flops are less frequency dependent. MSAFF gives the best low frequency yield, with 99%, while it's still poor for higher frequencies. SDFF is practically the same with a slightly lower yield at 1MHz. CTGFF gives a lower yield at lower frequencies and is almost the same for higher frequencies.

## 5.2 Voltage scaling yield

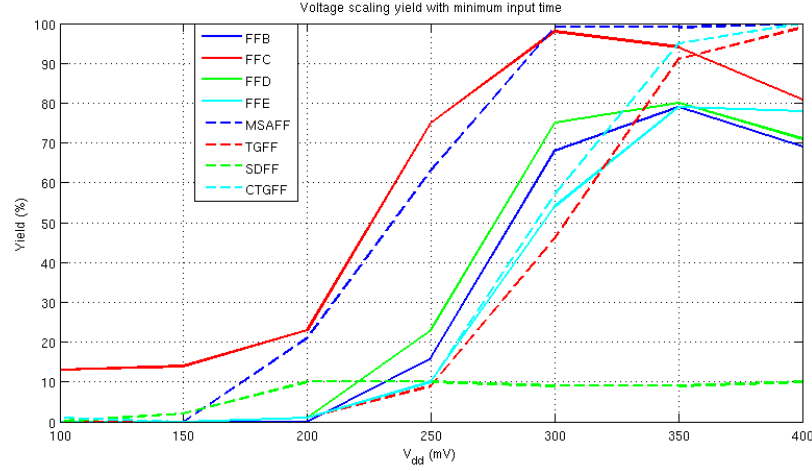


Figure 5.4: Plot showing the yield for all the flip-flops from  $V_{dd} = 100mV$  to  $400mV$  with minimum input time.

Figure 5.4 shows the yield for all flip-flops with varying supply voltage. The frequency has been set to  $100kHz$  for all flip-flops to make it equal for all of them. This frequency was chosen because it has the best overall yield and because several flip-flops don't work on higher frequencies. All supply voltages have the same restriction for switching time. This restriction has been set to  $2\mu s$  because the frequency is so low and it's enough for a supply voltage of  $100mV$  to be able to switch. All simulations have been done with minimum input time,  $t_{setup}$ . There is not much difference between the ULV flip-flops with the exception of FFC, which outperforms the others just as with the frequency yield. One thing to notice is that the yield drops at  $400mV$  for all the ULV flip-flops. FFC has its peak at  $300mV$  while the others peak at  $350mV$ . The reason for this is that the output signal switches too early in more cases at  $400mV$ . The time they switch early is equal to the input time, so by using the setup time as the input time they only switch  $< 2ns$  early. If one considers this to be negligible the yield would be 100% at  $400mV$  for FFC and the other ULV flip-flops would peak at this  $V_{dd}$  with minimum input time. The yield is presented this way because with a longer input time it's not negligible. This could be fixed by optimizing the dimensions of the transistors for each  $V_{dd}$ . This will not be done in this thesis as we base our thesis around  $300mV$ . FFC is also the only ULV flip-flop that gives any yield below  $200mV$ . The non-ULV flip-flops behave as expected with an increasing yield with increasing  $V_{dd}$ . All of them reach 100% yield at  $400mV$  except SDFF. SDFF did not work at  $100kHz$  and was not expected to give any good yield.

### 5.3 Delay variation

While looking at the yield it can also be interesting to see how the delay varies. Figure 5.5 shows a histogram of how the delay,  $t_{c-q}$ , varies relative to the expected delay. A high frequency of  $50MHz$  was chosen. The comparison flip-flops does not have sufficient yield at  $50MHz$  and thus are not included in this section. However the delay should not be dependent of frequency. The  $V_{dd}$  used was  $300mV$ . This way it's easy to compare it with earlier results from schematics (which will be the expected value), since the delay is very dependent on the supply voltage.

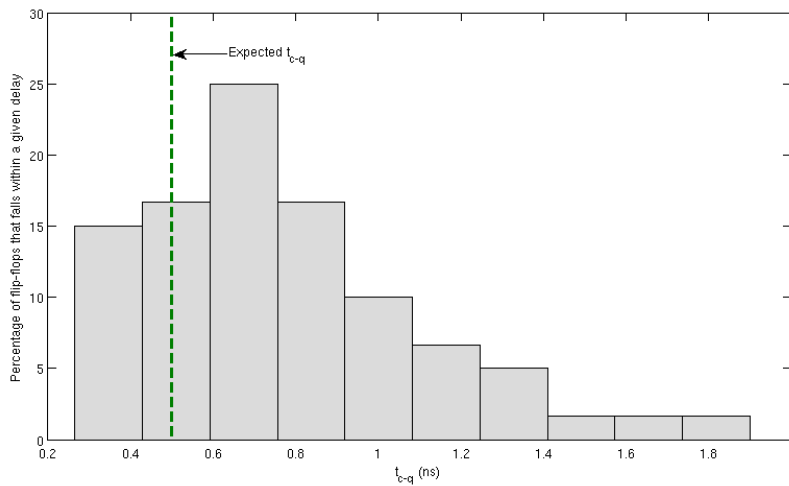


Figure 5.5: Histogram of FFB delay variation at  $50MHz$ .

As we can see from the figure it's more likely that the delay will be a little bit longer than the expected value. For FFB we can see that the probability peaks at around  $0.7ns$  while the expected value is  $0.5ns$ . The probability that the delay is less than expected is about 15%. The longest and shortest delay found was at around  $1.8ns$  and  $0.3ns$ . All of the failed outputs are excluded from all the histograms.

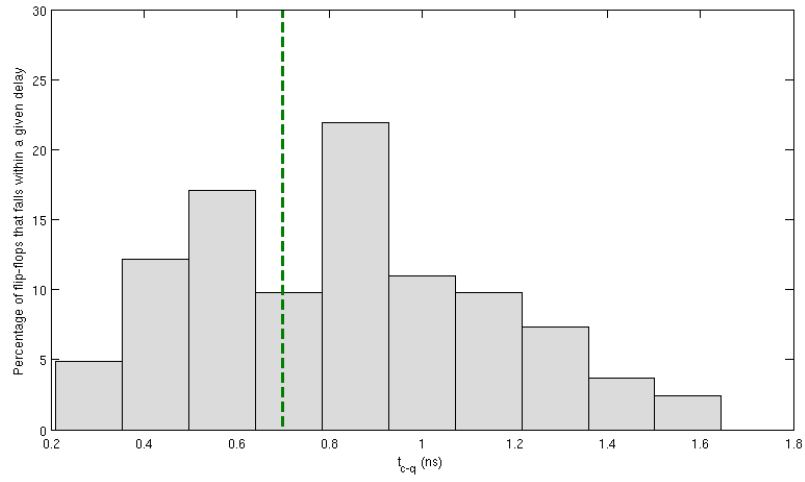


Figure 5.6: Histogram of FFC delay variation at 50MHz.

Figure 5.6 shows a histogram of FFC's delay variation. We can see that the delay peaks at around  $0.9ns$  while the expected value is  $0.7ns$ . There is also a 34% chance that the delay will be less than expected. The longest and shortest delay found was  $1.6ns$  and  $0.2ns$ .

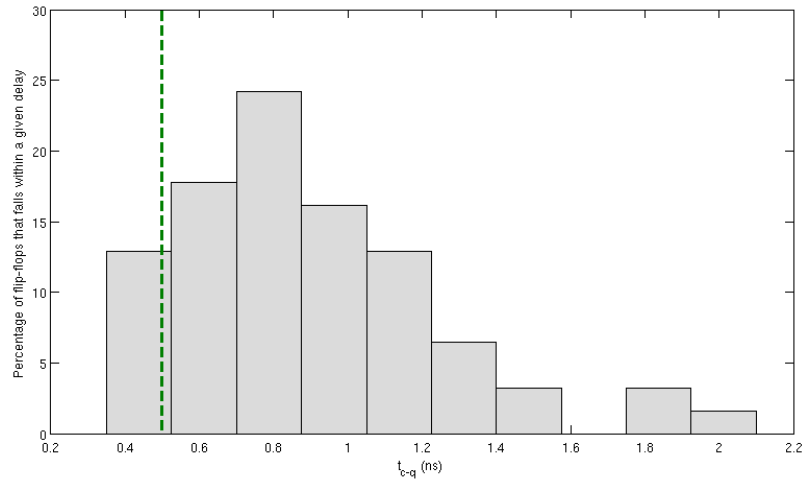


Figure 5.7: Histogram of FFD delay variation at 50MHz.

Figure 5.7 shows a histogram of the delay variation for FFD. The delay peaks at around  $0.8ns$  while the expected value is  $0.5ns$ . The longest and shortest delay was found to be  $2ns$  and  $0.4ns$ .

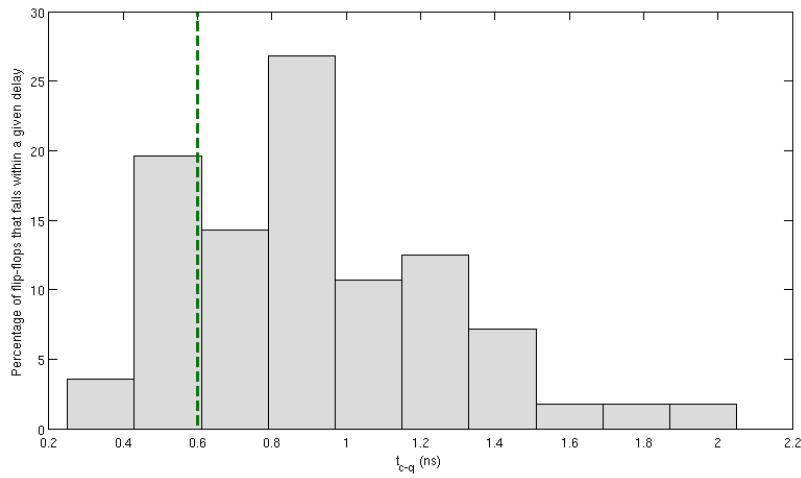


Figure 5.8: Histogram of FFE delay variation at 50MHz.

Figure 5.8 shows a histogram for FFE's delay variation. For FFE the delay peaks at around  $0.9ns$  with an expected value of  $0.6ns$ . There is a 24% probability that the delay will be less than expected. The longest and shortest delay was around  $2ns$  and  $0.3ns$ . The trend for all the flip-flops is that the longest likely delay is  $2 - 3ns$  longer than the expected delay.

After looking at all the results it's clear that the flip-flop which gives the most satisfying result is FFC. While the other ULV flip-flops perform similarly FFC outperform them considering frequency and supply voltage.





## Chapter 6

# Power and energy considerations

So far all focus has been on flip-flop speed by examining the delay and setup time for each flip-flop. This has been the primary focus so far because these flip-flops are designed specifically for high speed performance at low voltages[10]. However the energy consumption is also an interesting factor to look at. Of particular interest is the Energy Delay Product which scales with the delay, thus will take advantage of the high operation speed of the ULV flip-flops.

Power will be split into two different categories: dynamic power and static power.

The average power consumed during a flip-flop output switch will be referred to as the dynamic power of the flip-flop. This number can be used to determine the flip-flop Energy Delay Product independent of frequency to allow for a good comparison of the flip-flops. The data presented will be the average power only during the actual switching of the output signal, not the average power during a frequency dependent region of time. This means that the dynamic power as presented and defined in this thesis is expected to stay the same independent of frequency.

For stable output values where the flip-flop does not switch the average power consumption will be referred to as the static power consumption. Compared to dynamic power the static power consumption of a flip-flop is much lower, sometimes by more than an order of magnitude. This is because the flip-flop will spend much more energy switching it's output state compared to simply keeping it stable.

## 6.1 Power considerations at different frequencies

### 6.1.1 Static power

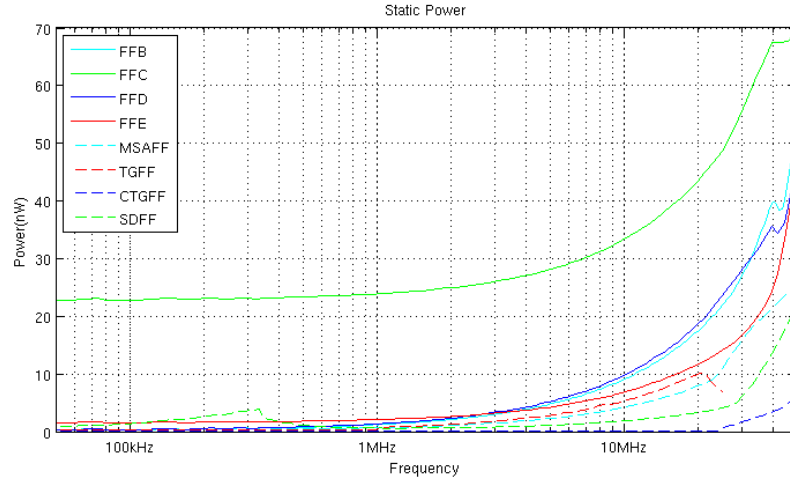


Figure 6.1: Average static power consumption for the different flip-flop topologies.

It is expected that at lower frequencies all flip-flops will spend less energy per time as a result of slower clock operation which reduces the amount of transistor activity going on. Indeed this is what we see from simulations sweeping the frequency in Figure 6.1. In order to keep all values comparable the input time remains unaltered for all frequencies and so does the supply voltage set at  $300mV$ . For actual implementation of the flip-flops supply voltage should be considered separately for each frequency in order to maximize delay response and energy consumption, however this would make the simulation process extremely cumbersome. Furthermore the explicit effect of frequency on the static power is better observed when all other conditions remains unaltered.

In addition a different strength relationship (strength of about 8–9, see Table 4.3) between the flip-flop inverters and recharge transistor were used in order to ensure that the flip-flops does not experience huge fluctuations on different frequencies as a result of the recharge transistor properties. This comes at the expense of the delay of the flip-flops, and this fact should be kept in mind.

In addition because of limitations to the accuracy of simulations in Cadence the delay is measured only on the falling edge of the flip-flop instead of taking the average of both. This will cause further deviations from previous delay results, however the difference should not be very significant.

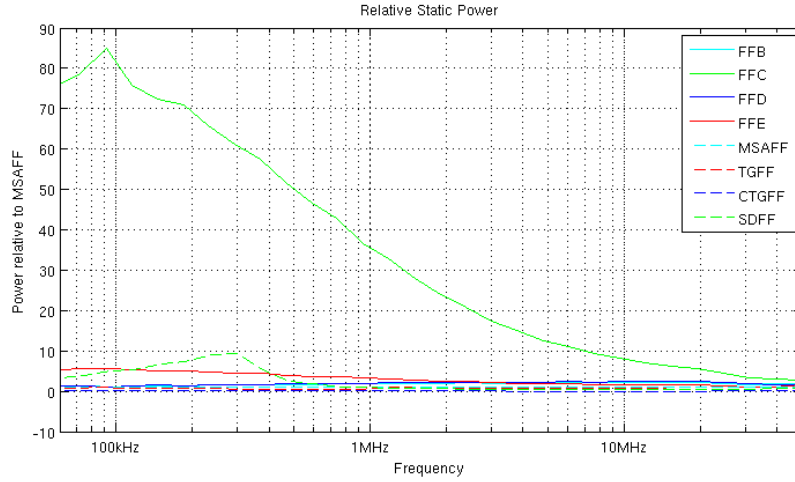


Figure 6.2: Average static power consumption normalized relative to the static power consumption of the MSAFF.

In Figure 6.2 the static power consumption has been normalized to better compare the different flip-flops at different frequencies. This normalization has been done relative to the MSAFF since this flip-flop is the comparison flip-flop. However it should be noted that other flip-flops score better on static power consumption notably the CTGFF.

It is clear from Figure 6.2 that FFC is very bad when it comes to static energy considerations. On this flip-flop the input signal  $D$  is connected through only 1 level of transistors to the output  $Q$ . The high energy consumption is likely a result of leakage from the input to the output making it much more difficult for the flip-flop to hold onto the output value. At the same time the output signal leaks back to the input thereby leaking into the output of the previous flip-flop. This further increases the energy needed to keep the output stable by the flip-flop inverters resulting in a high static power consumption. Overall this design should not be considered if static energy represents a large chunk of the total energy used (for circuits with frequent switching of flip-flop values).

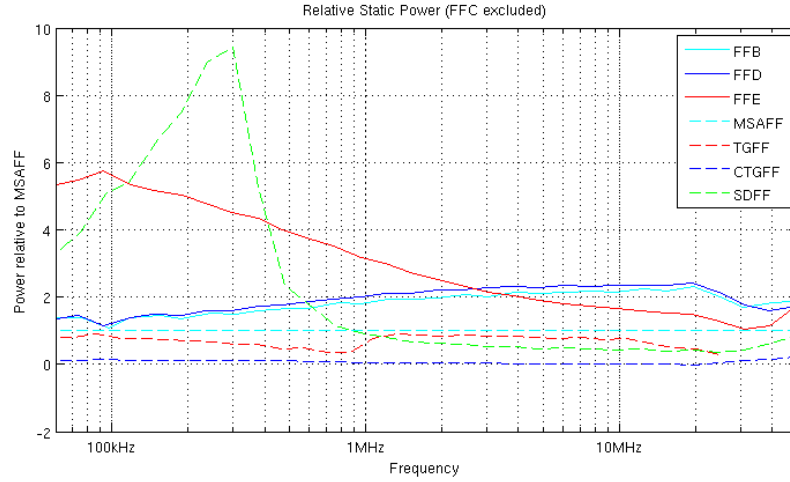


Figure 6.3: Average static power consumption normalized relative to the static power consumption of the MSAFF.

In Figure 6.3 FFC has been removed in order to better view the results. As mentioned earlier the conventional flip-flop scores extremely well on static power consumption. Once it has latched onto an input signal there is virtually no leakage and the output remains extremely stable on its own. However as has been established previously this flip-flop also suffers from very bad delay properties which makes it unsuitable for high speed application which is the main focus of this thesis.

Furthermore the MSAFF is in fact the worst out of the four comparison flip-flops for most frequencies, while in the sub- $MHz$  region the SDFF takes over as worst flip-flop for static power considerations. The fact that the MSAFF does not perform that well in power considerations is the reason why all comparison flip-flops are included in this analysis.

In general the ULV flip-flops perform worse than the other flip-flops in static power consumption. This is due to the capacitors integrated in the flip-flop design. Extra energy is lost when these capacitors are charged and then emptied during each clock cycle. However the overall loss of energy is not extreme if one considers that static power will not be the dominant power consumption during normal operation of flip-flops.

All flip-flops mostly remains within the same order of magnitude in static power consumption, however there are deviations from this at the sub- $MHz$  frequencies. The SDFF has already been mentioned, and the second deviation is FFE. This is the flip-flop of which the keepers are controlled by an output feedback instead of being clocked. As the frequency decreases the extra load on the output from these inverters requires comparatively more and more energy in order for the flip-flop inverters to keep the output stable.

It should be kept in mind that static energy consumption is less than dynamic energy consumption so these considerations should be focused on only if the flip-flop usage causes the static power to be much more important. This will be the case for circuitry that holds onto the output

signal for a long duration of time without latching onto anything new.

### 6.1.2 Dynamic power, delay and EDP

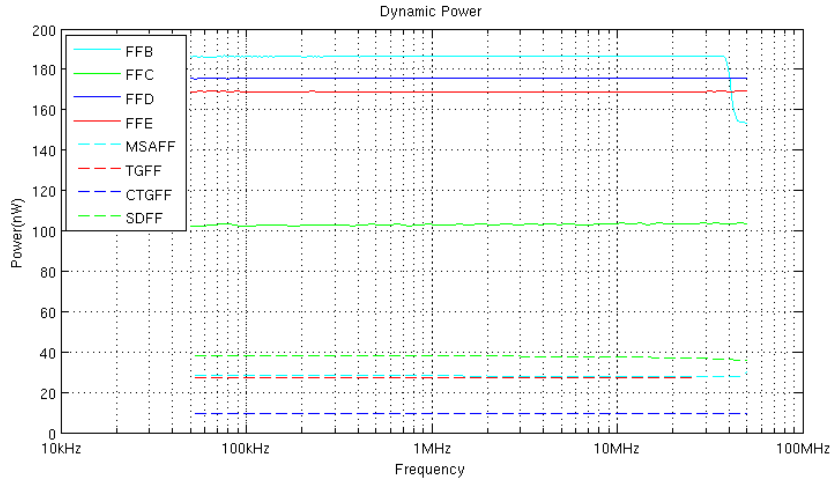


Figure 6.4: Average dynamic power consumption for the different flip-flop topologies.

Because the dynamic power represents only the power used during a single switching event it is expected to be independent of operating frequency. As can be seen in Figure 6.4 this is indeed the case. The small decrease in dynamic power seen on FFB at the highest frequencies is likely a measuring error rather an artifact of the flip-flop properties. In any case all ULV flip-flops show significantly more dynamic energy consumption than any of the comparison flip-flops.

Because of the semi-floating gates the ULV flip-flops experience an extra boost in transistor current for the evaluate transistors which naturally will cause the dynamic energy consumption to increase dramatically compared to flip-flop topologies that does not use this kind of method.

While it was observed previously that FFC spent significantly more energy in during static periods it can be seen in Figure 6.4 that the reverse is true for the dynamic period. During switching FFC spends about half as much energy as the other ULV flip-flops. However it still consumes significantly more power than the comparison flip-flops.

Also notice that among the comparison flip-flops the winner is again found to be the CTGFF is the least power consuming one by a large margin.

Furthermore one can see that dynamic power consumption is about one order of magnitude higher than static power consumption, making it a much more significant contributor to the total amount of energy spent. This is especially true for circuits that experience frequent latching of new values in the flip-flops.

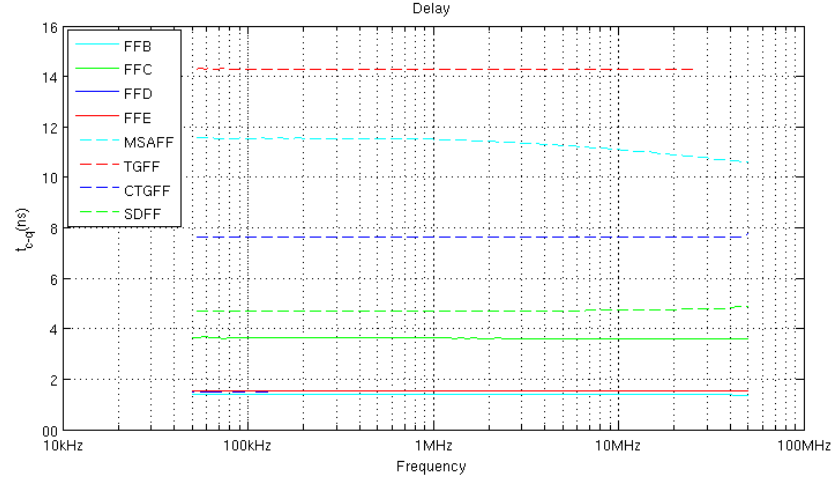


Figure 6.5: Delay  $t_{c-q}$  for the different flip-flop topologies as a function of frequency.

As with the dynamic power consumption it is expected that the delay will remain mostly unaffected by the operating frequency of the circuit. However one can see in Figure 6.5 that there is a slow deviation in the delay of the MSAFF from 1MHz to 50MHz. The cause of this deviation is not known, however the purpose of this thesis is not to study the MSAFF topology in detail. The magnitude of the deviation is not concerning as it only fluctuates with about 15% and does not have a huge impact on how good this flip-flop is compared to the others.

Figure 6.5 confirms the previous examinations of delay in regard to which flip-flops are faster. However the FFC has suffered much more than the other ULV flip-flops in delay. It still remains the slowest of the ULV flip-flops.

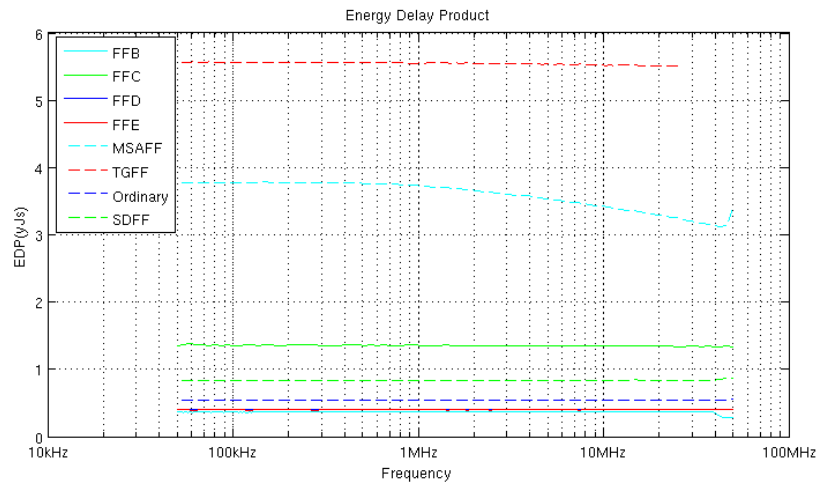


Figure 6.6: EDP for the different flip-flop topologies as a function of frequency.

Using the data from Figure 6.4 and 6.5 EDP can be evaluated as seen in Figure 6.6. With the two previous data sets being mostly independent of frequency, so is the EDP.

It can be seen that while all ULV flip-flops had high power consumption the EDP of almost all ULV flip-flops are significantly better than the comparison flip-flops. The exception is again FFC which proves to have a much worse EDP than the other ULV flip-flops due to its longer delay at these inverter strength/evaluate transistor strength values.

The TGFF and MSAFF scores extremely bad in EDP with much higher values than all other flip-flops. The CTGFF and SDFF actually score better than FFC on EDP while still having values higher than the rest of the ULV flip-flops. CTGFF does however come close to the EDP values of the ULV flip-flops.

Overall the ULV flip-flops have a very good EDP compared to the other flip-flop topologies. This shows that the ULV flip-flops are usable even if power is a concern for the circuit design over a large frequency spectrum.

It should be noted that there are other ways to weight the importance of power vs delay by using different exponential values for P and D as explained in [33].

## **6.2 Power considerations at different supply voltages**

As has been shown previously the static power varies depending on the operating frequency. Furthermore at low voltage values the flip-flops will have a larger delay so the delay and frequency limitations vary depending on the supply voltage. An effective way to ensure that the flip-flop simulations will run properly is then to lower the operating frequency for lower supply voltages. This however will distort the static power result as it is dependent on both the supply voltage and frequency which are now both being swept.

In order to be able to look exclusively at the effect of supply voltage on the static power of the flip-flops a different approach has to be taken. Instead of sweeping the frequency with the supply voltage, we will use a low frequency to ensure operational flip-flops even at low supply voltage. For this a frequency of  $10kHz$  was chosen as this is one of the lowest frequencies that can be used.

There were limitations to the Cadence simulation tool and MATLAB as explained in Appendix B. The way this problem has been solved in our examinations is by splitting up the simulation into two parts. One simulation looks at the static power and does not need to look at the latching period or delay. This simulation then does not need to have such small time steps and can instead use a large time step to lower the simulation resolution. The second simulation looks only at the latching period which heavily reduces the total simulation time that needs to be imported into matlab. This second simulation can then be used to calculate

delay and dynamic power with a small enough time step that the data can be trusted.

This unfortunately proves to be a problem for some of the flip-flop topologies. The SDFF generally experience more problems at lower frequencies, and especially with the high input time that has to be used in order to satisfy low supply voltages, this flip-flop does not yield any acceptable results for this test. As such it has been excluded.

Additionally FFC is the only ULV flip-flop that suffers from hold time issues. This becomes extremely problematic at high input time which is needed for the lower supply voltages of this simulation. As with the SDFF, the FFC has been excluded from these results because the results gained were not in any way usable.

For both of these flip-flops it is recommended that the reader look at the frequency sweeps to evaluate power and energy concerns.

It should also be noted that the matlab script that handles the calculations inserts a value of 0 at supply voltages that are too low for a flip-flop to work. The zeros seen on the figures are not to be assumed as actual values and should be ignored.

Keep also in mind that there will be small divergencies from the frequency sweep because of the altered simulation conditions that were needed in order to get working results. However these divergencies are generally not very big.

### 6.2.1 Static power

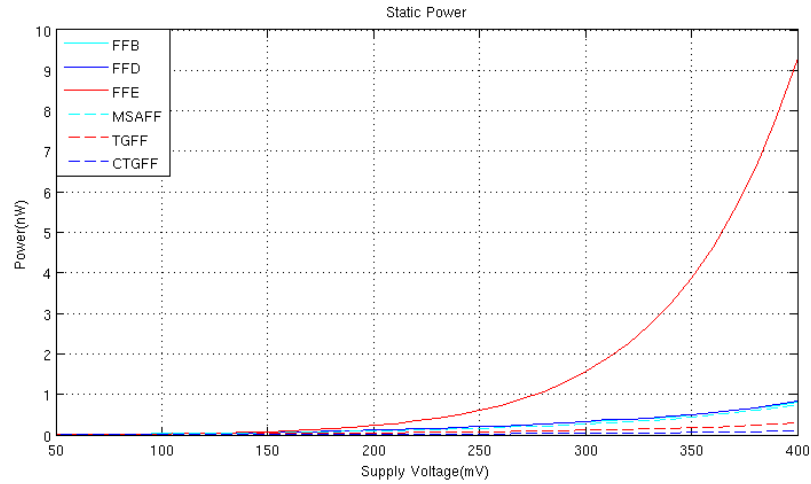


Figure 6.7: Average static power consumption for the different flip-flop topologies at different supply voltages.



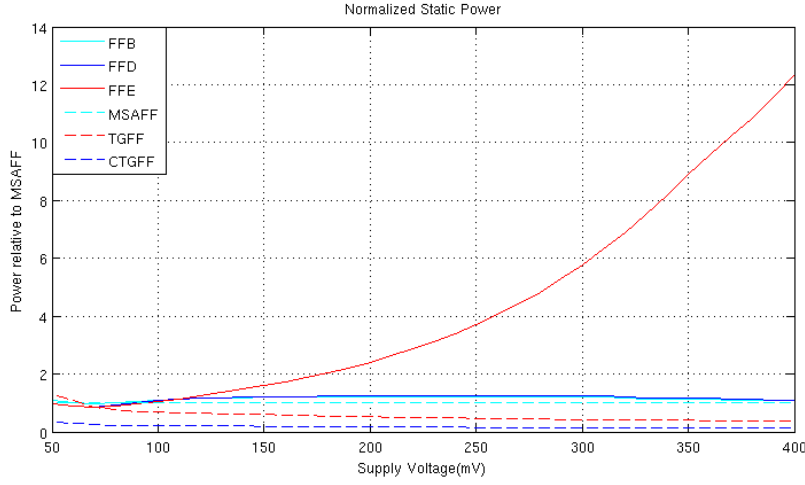


Figure 6.8: Average static power consumption for the different flip-flop topologies at different supply voltages relative to the MSAFF.

On figures 6.7 and 6.8 the static power's relationship to the supply voltage can be seen. The first figure shows the actual values while the second figure shows normalized values which makes it easier to compare the flip-flops at different voltages.

As expected the power consumption decreases rapidly with the supply voltage. Generally reducing the supply voltage is a way to decrease power consumption [2].

As seen in Figure 6.7 the FFE is the flip-flop with the highest static power consumption. It's power consumption is significantly higher than all other flip-flops. However taking a look at Figure 6.8 reveals that the static power consumption of FFE approaches the values of the other flip-flops as the supply voltage gets lower, finally catching up by the time  $100mV$  is reached.

As has been discussed in an earlier section the cause of FFE's much higher static power consumption is a result of the keeper transistor gates that has to be driven by the output of the flip-flop, hence adding additional static power consumption.

The ULV flip-flops in general use more static power than the other flip-flops, and the relationship between the power consumption of the flip-flops stays mostly the same for all supply voltages with the exception of FFE. The previous trend of CTGFF is also apparent in these simulations where again it's the least power consuming out of all the flip-flops.

## 6.2.2 Dynamic power

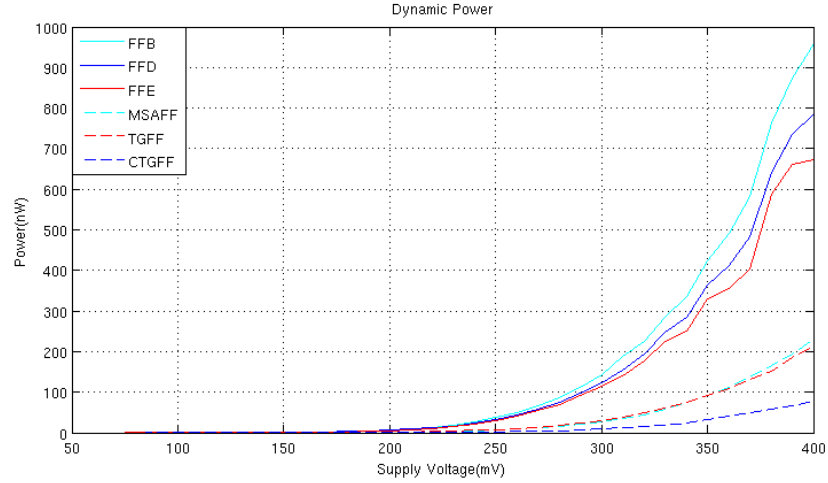


Figure 6.9: Average dynamic power consumption for the different flip-flop topologies at different supply voltages.

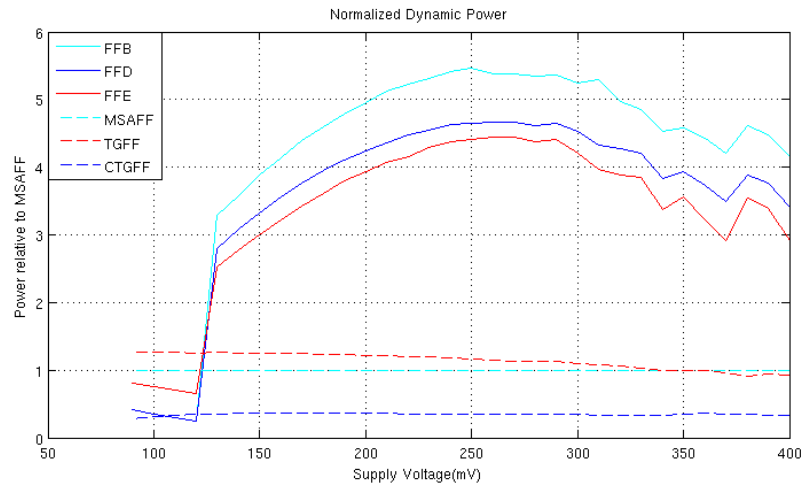


Figure 6.10: Average dynamic power consumption for the different flip-flop topologies at different supply voltages relative to the MSAFF.

In Figures 6.9 and 6.10 the Dynamic power consumption of the flip-flops can be seen as a function of supply voltage. The ULV flip-flops are much more dominant here compared to static power consumption because as explained in an earlier section, these flip-flops operate at a high speed as a result of much more significant transistor current which leads to an overall higher power consumption during latching.

Looking at Figure 6.10 the normalized power consumption can be analyzed.

As per usual the CTGFF shows excellent power properties and is far

superior to all other flip-flops in this regard.

The flip-flops seem to generally have the same relative dynamic power consumption to each other, however there is much more of a variation in the ULV flip-flops. At about  $130mV$  the ULV flip-flops seem to suddenly drop below the dynamic power values of the other flip-flops. The cause of this sudden change in pattern is not clear and needs to be investigated further in the future. For now let's only note that the relative dynamic power consumption of the ULV flip-flops seems to peak at about  $150mV$  and decrease much more rapidly before this point than after.

Both the TGFF and MSAFF seem to be very similar in dynamic power qualities at all supply voltages, and the ULV flip-flops behave extremely similar with a distance between each other that is kept constant regardless of voltage.

Overall the ULV flip-flops seem to use roughly five times as much power during a latching than the comparison flip-flops.

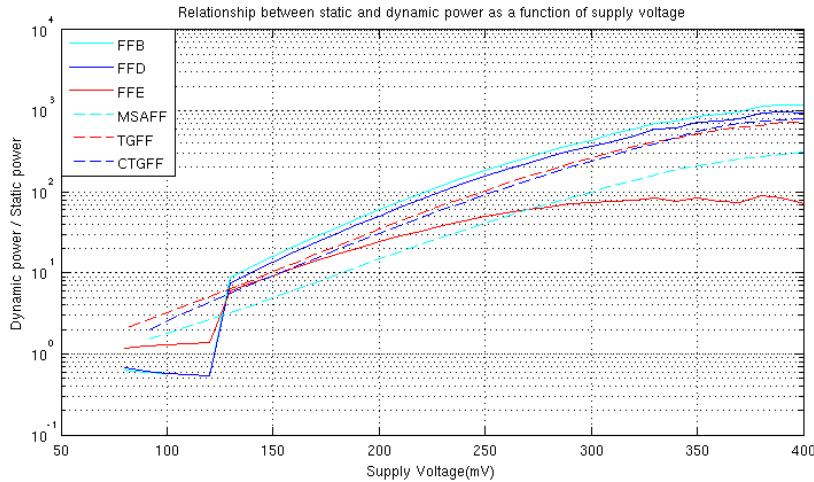


Figure 6.11: Dynamic power compared to static power for the different flip-flops at different supply voltages.

While previously the dynamic power has been shown to be about an order higher than static power, it's not as simple for these simulations. Previously it was shown that static power decreases with the frequency, and because a low frequency was chosen in order to do these tests the static power for higher supply voltages is much lower. This causes there to be a difference in two to three orders between static and dynamic power for high voltages.

It can be seen in Figure 6.11 that the relationship between the two is dependent on supply voltage. As the voltage decreases the static power consumption becomes more and more dominant. This makes much sense since much of the static power consumption is a result of leakage which dominates on lower voltages [referer alioto]. It can also be seen as a similar situation as  $I_{on}/I_{off}$ , where the transistor activity is averagely low in static power and very high in dynamic power. As the supply voltage gets

lower the  $I_{on}/I_{off}$  becomes lower as well as the  $I_{on}$  current approaches the  $I_{off}$  current until there is little difference between transistors being off or on[25].

As the supply voltage reaches about  $100mV$  the static and dynamic power consumption are within the same order of magnitude.

It is therefore important to consider what supply voltage will be used if power considerations are to be made. On high voltages the effect of static power will be very minimal, while on low supply voltages static power considerations needs to be taken into account, and might even outweigh the dynamic considerations if the conditions are right.

### 6.2.3 Delay

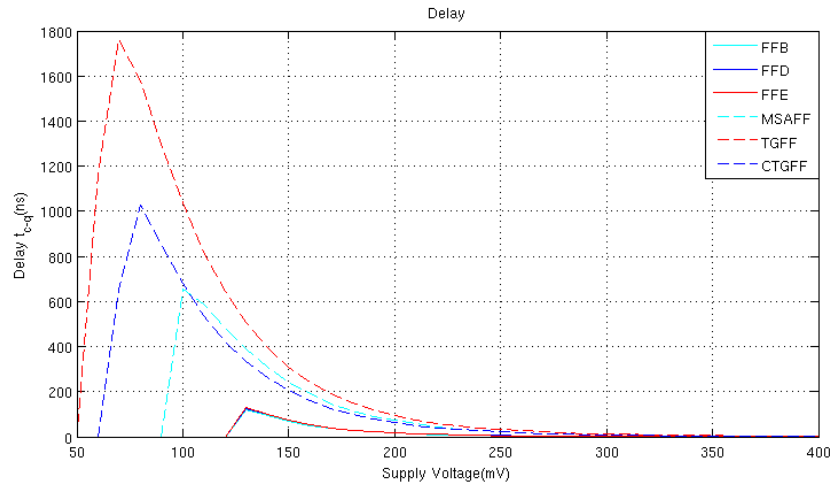


Figure 6.12: Delay for the different flip-flop topologies at different supply voltages.

In Figure 6.12 the delay as a function of supply voltage for the different flip-flops is shown. The general tendency for flip-flops delay[34][35] is seen to be present on the ULV flip-flops as well as all the comparison flip-flops. The delay increases dramatically as the supply voltage gets lower, which heavily reduces the maximum operational frequency for the flip-flops.

However as can be seen the ULV flip-flops maintains a much better delay at all supply voltages. There seems to be no flip-flop in particular that stands out on this graph. There is however a clear grouping between the ULV flip-flops and comparison flip-flops which seems to behave similarly to each other.

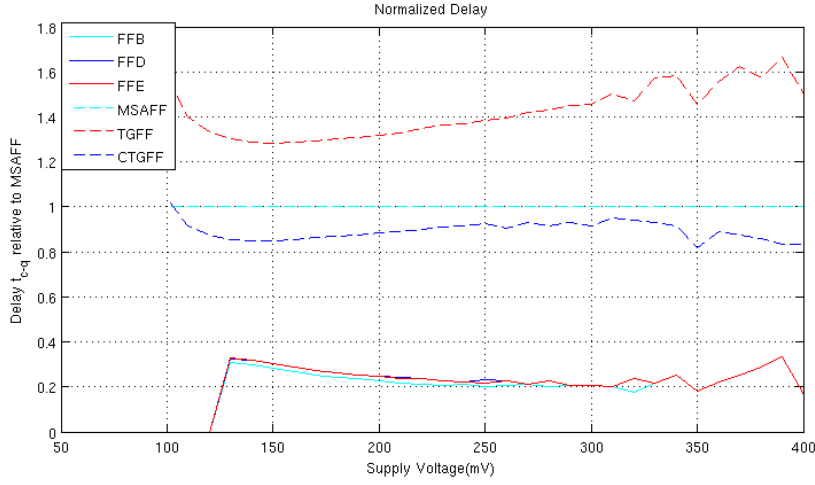


Figure 6.13: Delay for the different flip-flop topologies at different supply voltages relative to the MSAFF.

In Figure 6.13 normalized delay is shown. The results show that the ULV flip-flops are about 5 – 10 times faster on all supply voltages. Furthermore the difference between the ULV flip-flops is never big in comparison to the rest, so for delay considerations alone it is not all that important which ULV flip-flop is used regardless of supply voltage.

The delay behaves as expected in relation to the supply voltage. The main purpose of this sweep is not to look at the delay but to be able to calculate the energy delay product.

#### 6.2.4 Energy Delay Product

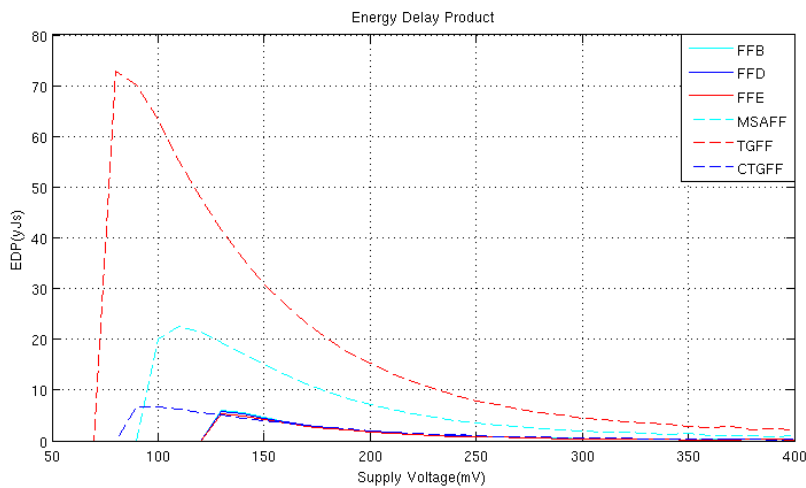


Figure 6.14: EDP for the different flip-flop topologies at different supply voltages.

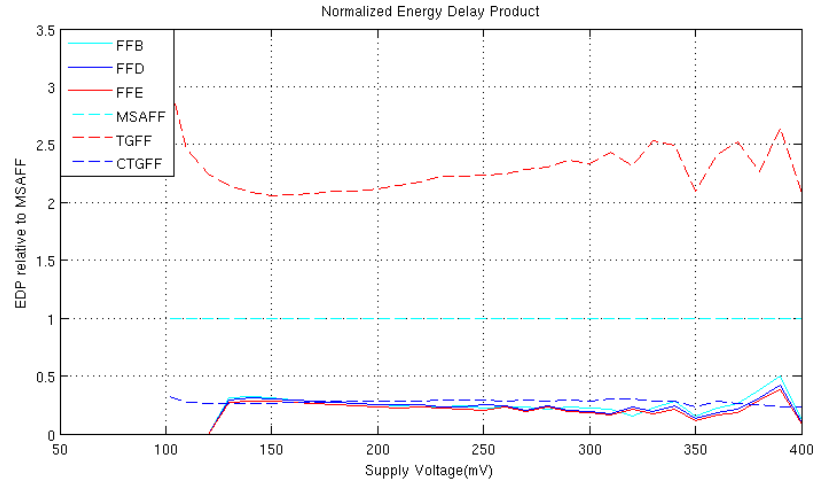


Figure 6.15: EDP for the different flip-flop topologies at different supply voltages relative to the MSAFF.

In Figures 6.14 and 6.15 the EDP is finally calculated based on the previous results. Because EDP scales with delay squared the figure takes a shape similar to the delay since the effect of power is smaller on this parameter.

Both the TGFF and MSAFF stands out as the worse flip-flops in these simulations. The TGFF in particular does not possess a very good EDP.

The CTGFF on the other hand proves to have an EDP almost equal to that of the ULV flip-flops which all have a very good EDP on all supply voltages. At very low supply voltages it may look like the CTGFF have a small advantage over the ULV flip-flops, however the difference as seen in this simulation is not big enough to determine whether it would actually have a lower EDP if tested on actual circuitry. In general it can be assumed that the ULV flip-flops and CTGFF are equally good when making considerations of EDP across all supply voltages.

These trends appear regardless of the supply voltage as can be seen in Figure 6.15, and the relative EDP between the flip-flops remains very stable.

Overall these simulations have shown that the ULV flip-flops can hold up against other flip-flop typologies for EDP considerations. This means that even if power is a factor in the design of a circuit the ULV flip-flops can be considered as long as operational speed is also important. If speed is completely irrelevant then the ULV flip-flops does not necessarily weigh up for their higher power consumptions. However in circuitry where some parts needs fast operation the ULV flip-flops can be used without necessarily draining too much power compared to the rest of the circuitry, especially if only a small part of the total circuitry needs to operate at a high speed.

## Chapter 7

# Comparison between the ULV flip-flops

Flip-flop	$t_{setup}$	$t_{c-q}$	Yield	EDP
FFB	+1.9ns	20.0× faster	7.0×	0.12×
FFC	+1.0ns	14.3× faster	14.7×	0.38×
FFD	+1.8ns	20.0× faster	8.0×	0.12×
FFE	+1.2ns	16.7× faster	4.8×	0.12×

Table 7.1: Comparison between improvement relative to the reference flip-flop: MSAFF. Yield is given at 300mV and 10MHz.

In Table 7.1 a compact comparison between the different ULV flip-flops is shown. It's immediately clear that the ULV flip-flops are significantly better than the MSAFF. The small added setup time is a small price to pay for the large gain in speed, yield and lower EDP. In addition to the stats in Table 7.1 all of the ULV flip-flops retains a -38% area advantage over the MSAFF.

Keep in mind that yield qualities are highly dependent on operating frequency and supply voltage, so the yield improvement number given can vary a lot depending on the environment the flip-flops will be implemented into.

### 7.1 FFB

This flip-flop together with FFE have the fastest available switching time with a delay of only 0.5ns, making it 20 times as fast as the MSAFF. This flip-flop also has a very simple and straightforward input where the clock and  $D$  signals are not sent through the keeper or evaluate transistors. This gives low input load, adds no load to the output and also does not add any extra load for the clock. However this comes at the cost of a very high setup time; The longest setup time of any ULV flip-flop at 1.9ns.

## 7.2 FFC

FFC has the best yield of all the ULV flip-flops. It has a yield that is around twice as good as for example FFB and FFD, and almost 15 times as good as the MSAFF. This can be a very significant factor in choosing a flip-flop for implementation in a specific design. For any design that employs FFC this will probably be the main reason. In addition this flip-flop has the shortest setup time among the ULV flip-flops as well as the lowest dynamic power consumption. Finally the spread of delay variation on FFC yields lower delay than that of other ULV flip-flops, making it an even better choice for high yield considerations.

However these advantages comes at a high cost. FFC has the lowest switching time of all ULV flip-flops with a delay of  $0.7ns$ . However this is only  $0.2ns$  more than the fastest ULV flip-flop, so this is not necessarily a problem considering the high yield. Of greater concern is the added hold time of  $4.0ns$  which is not present in other ULV flip-flops. This puts extra restrictions on design requirements and might be extra disadvantageous for some types of circuits. Furthermore the EDP is about three times as high as the other ULV flip-flops, although this is still significantly less than that of the comparison flip-flop. The biggest drawback of FFC could be the high static power consumption which makes FFC particularly unsuited for low frequency environments. FFC also suffers from a slight increase in the input load, which has to be taken into consideration if this flip-flop is to be implemented.

## 7.3 FFD

This flip-flop shares the fastest switching time with FFB with a delay of  $0.5ns$ . However it retains a small advantage in setup time compared to FFB with a setup time of  $1.8ns$  which is  $0.1ns$  faster. This is not much however, and the setup time is the second slowest among the ULV flip-flops. In addition FFD has the second highest yield output, although FFC is better in this regard with a large margin. The yield is only a little better than that of FFB. Overall this flip-flop is very similar to FFB with small differences.

## 7.4 FFE

FFE has a very short setup time of only  $1.2ns$  which is only  $0.2ns$  slower than that of FFC which has the fastest setup time. It's keeper configuration puts much less load on the clock and allows the keepers to operate in a more intuitive way, independent of the clock. For yield considerations this flip-flop stands out at low frequencies and high input time. In these conditions FFE retains a respectable yield where the other ULV flip-flops fails to yield any working circuits at all. However at other frequencies this flip-flop has the worst yield out of all the ULV flip-flops. Furthermore this flip-flop has poor static power consumption properties. This is only a problem for high



supply voltages or low frequencies, and is much less of a problem at low supply voltages and high frequencies.



## Chapter 8

# Layout and chip design

For the chip design we had to choose one of the several flip-flops. We ended up choosing FFD. The reason we chose this was because it was the fastest alongside FFB, but with slightly less setup time as can be seen from 7.1. This decision was made at an early stage of the thesis where yield and EDP considerations had not been done yet. It's important to start with layout and chip design early on because it is time consuming and the manufacturing of the chip takes time.

For the layout design it was decided to implement the transistors with floating bulk. Floating bulk is the most extensive implementation for bulk biasing. It will be shown that it's possible to implement this for an ULV flip-flop, suggesting that most bulk implementations should work. Other bulk implementations as those seen in the schematics offer more efficient solutions, and are recommended for further work.

### 8.1 Layout

#### 8.1.1 Deep n-well implementation for floating bulk

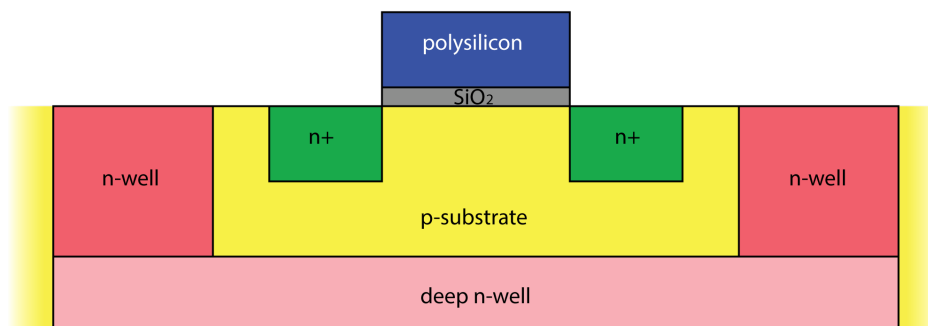


Figure 8.1: Cross section of a generic deep n-well implementation.

In order to isolate the bulk thus allowing for floating bulk transistors, deep n-well structures were implemented in the layout design for nMOS

transistors. In addition the nMOS transistors were enclosed in n-well to completely seal them off. This separated the p-well from all other p-wells, making it possible to have individual floating bulks for each transistor as seen in Figure 8.1. For the pMOS transistors this was not necessary since the n-wells are already separated from each other. The implementation of deep n-wells caused the minimum distance between transistors to increase dramatically because of restrictions on distance between deep n-well areas of different voltage potential. Figure 8.2 shows the implementation of deep n-wells in layout. The red color is the n-well and the light red color is the deep n-well. As one can see from the cross section (Figure 8.1) and the layout, the p-substrate for the nMOS is completely enclosed in n-substrate on all sides and underneath the nMOS.

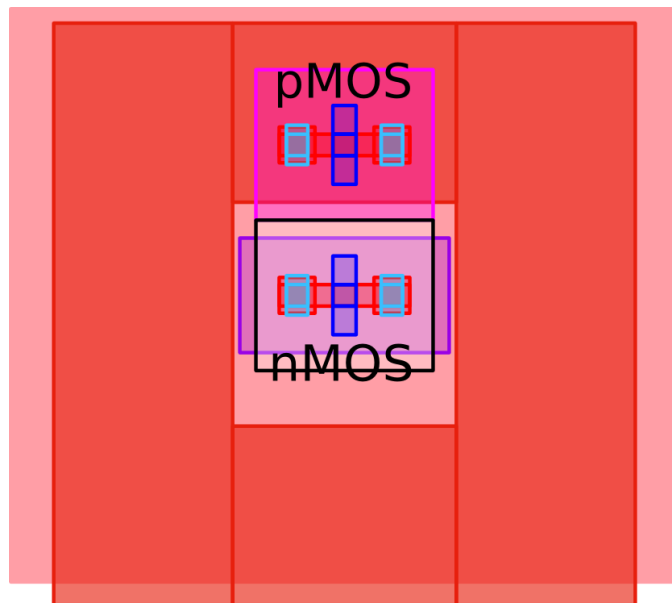


Figure 8.2: Layout of a pair of transistors implemented with deep n-well.

### 8.1.2 Crosstalk as floating gate capacitance

CMOS capacitors can be implemented in several ways. They can be implemented by using transistors which can be done schematically or by using a layout implementation of conductor-insulator-conductor.

One way to implement capacitors in CMOS is by using a transistor modeled as a lumped-RC component [36] in order to achieve a MOS gate capacitor. This can be seen on 8.3. Capacitors can also be implemented as PN junction capacitors. Both these methods use transistors in order to achieve a capacitor implementation.

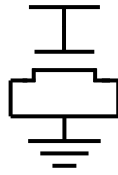


Figure 8.3: Transistor connected to be modeled as a lumped-RC component.

A common way to implement conductor-insulator-conductor capacitors is by using polysilicon-oxide-polysilicon (poly-poly) capacitors. In this variation the capacitance is obtained by two pieces of polysilicon separated by an oxide layer. However in many CMOS processes there are limitations to where the poly layers can be placed which limits the flexibility of the layout when implementing these kinds of capacitors.

For the layout of this thesis the capacitors will be implemented using metal-insulator-metal (MIM) capacitors or crosstalk capacitors. These capacitors are made using the metal layers of the CMOS process, allowing for many options on where to place the capacitors and what shape they can have. For example entire unused metal layers can be used to implement capacitors, or metal layers that are only used for a low number of wires. This way chip area can be saved. This kind of capacitor is suitable for the small capacitive values that are required in the ULV flip-flops. Small or large strips of metal can be used, allowing for capacitive values that are often lower than that of other capacitor types.

Crosstalk between different metal wires in the layout can often be a problem. Instead of solving the problem by spacing wires far apart, one can take advantage of the crosstalk in order to achieve wanted capacitive connections in the layout. This requires some clever thinking from a design perspective and might not always be possible. For this thesis it will merely be shown that MIM capacitors can be implemented effectively for ULV flip-flops. The implementation of floating bulk already increases the area usage of the circuit to a degree where it is difficult to save space with clever MIM capacitor configurations.

There are several possible ways to implement capacitors using the metal layers. Using two different metal layers for capacitive connections proved to be less effective than using capacitance between two metal strips in the same layer. This is because the distance between the layers are long compared to the minimum distance between two metal strips in the same layer. The most practical structural solution was to use “fingers”. Figure 8.4 shows an illustration of what MIM capacitors look like in the layout using “fingers”.

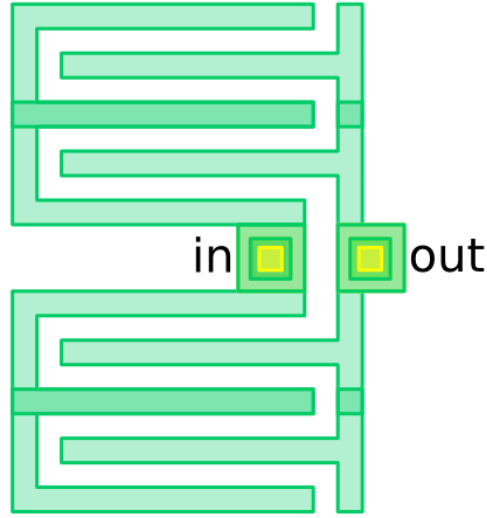


Figure 8.4: An illustration of MIM capacitors in layout. The yellow rectangle is a contact between metal layer 2 and 3, where yellow is layer 2 and green is layer 3.

The capacitive value between two wires can be calculated using the formula:

$$C = \epsilon_r \epsilon_0 \frac{A}{d} \quad (8.1)$$

Where  $\epsilon_r$  is the dielectric constant which is a material dependent value, and  $\epsilon_0$  is the permittivity of free space which equals:

$$\epsilon_0 = 8.854187817 pFm^{-1}$$

$A$  is the surface area of the capacitor, which in this capacitor layout is equivalent to the area of the vertical side. The distance  $d$  is the distance between the two wires, which we want to make as short as possible since this will give more capacitance per area on the chip layout.

It was decided to use metal layer 3 which had the following properties:

Minimum distance between wires :	$d_{min} = 140nm$
Layer thickness $T$ · length of wire $L$ :	$A = 310nm \cdot L$
Material between metal wires :	$\epsilon_r = 2.9$

The wire length  $L$  that is needed in order to get  $1fF$  is given by:

$$L = \frac{Cd}{T\epsilon_r\epsilon_0} = 17.6\mu m$$

In addition to the capacitance between the wires, there will be a little capacitive value added from the layer contacts between metal layers. So in order to figure out the size needed to get the wanted capacitor values simple

testing between two lines in layer 3 was done. Testing on schematical simulations showed that values of several  $fF$  was too high to let the flip-flops work at the wanted clock frequency. Simulations on just a single flip-flop showed that values as low as  $1aF$  would still be enough for the flip-flop to work, however later schematical simulations using several flip-flops showed that problems with cascading are present for very low values of  $C$ . Because of this it's important to try to stay as close to  $1fF$  as possible to give some headroom in both directions.

### **8.1.3 Final flip-flop layout**

The final layout of the flip-flop can be seen in Figure 8.5. As one can see the flip-flop takes up a much larger area than it could have because of the floating bulk implementation. All the n-wells and deep n-wells are minimum distance to each other. It's also interesting to note that the crosstalk capacitance (shown in green) takes up little space in one metal layer. Since it only uses one metal layer it's simple as well as practical to implement.

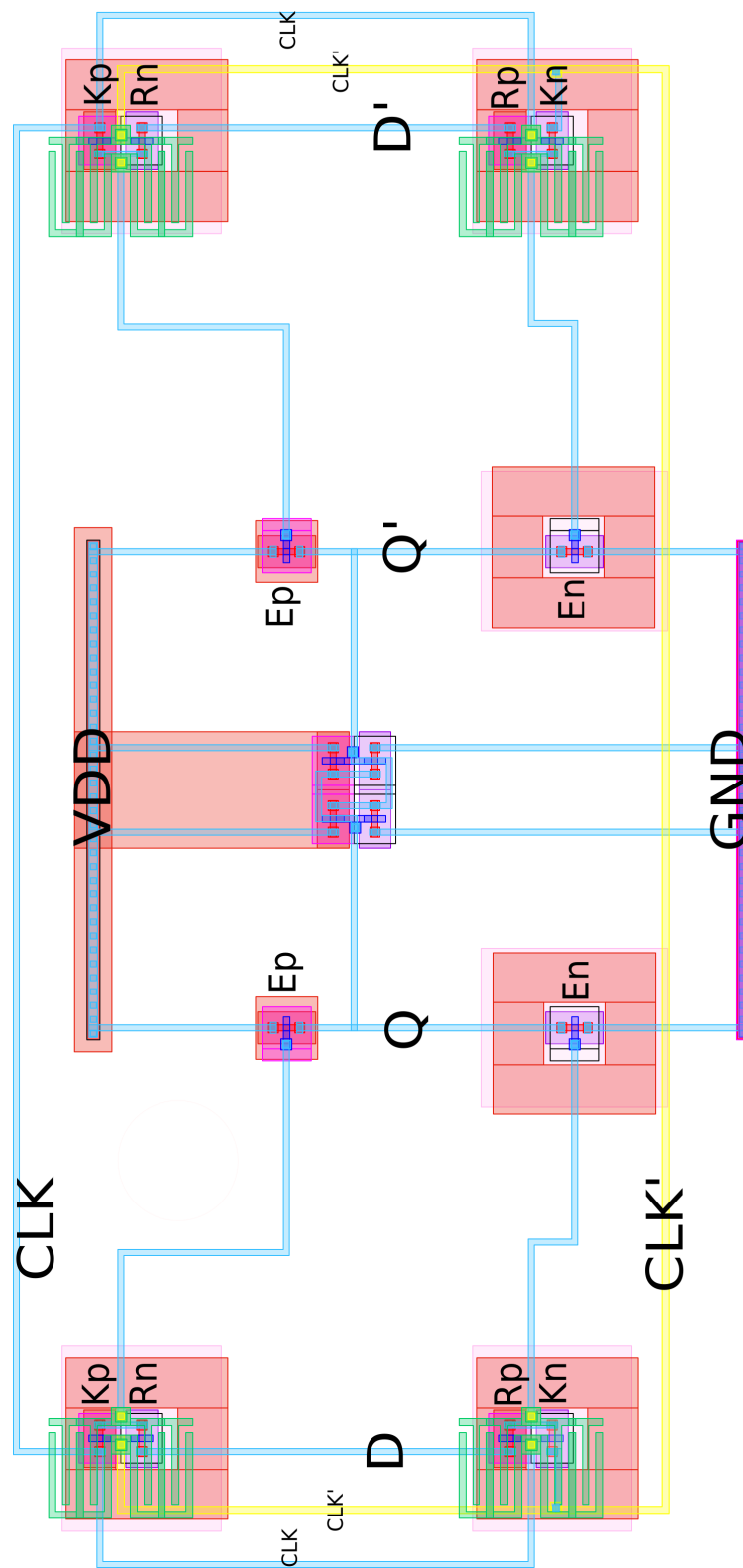


Figure 8.5: Final layout of the flip-flop



### 8.1.4 Amplifier

In order to do measurements on the chip, it needs to be able to handle a certain amount of output load. Since the ULV flip-flops can't handle this kind of load by themselves, there is a need for an amplifier on the output signal. The load on the pads and the measuring equipment was estimated to be around  $10pF$  by the engineer responsible for chip production. The amplifier we chose is a two-stage OP amp which can be found in [37]. This amplifier was chosen because it can drive a load of  $10pF$ . The  $V_{out}$  signal is connected to the  $V_-$  input signal so that we get a closed loop feedback that results in a voltage follower. It's set up as a voltage follower because we want it to work as an analog buffer.

The ULV flip-flop is too fast for the amplifier to give a satisfying analog output at  $300mV$ . Therefore the amplifier is set to operate with a supply voltage of  $1.2V$  so it's capable of making fast enough rise and fall transitions. The ground signal is to set  $-500mV$  and the  $V_{dd}$  is set to  $700mV$  relative to the ground signal on the flip-flop. Schematic of the amplifier can be seen in Figure 8.6 and the layout can be seen in Figure 8.7. As one can see from the layout, the amplifier uses deep n-well implementation to get separate body bias from the flip-flop.

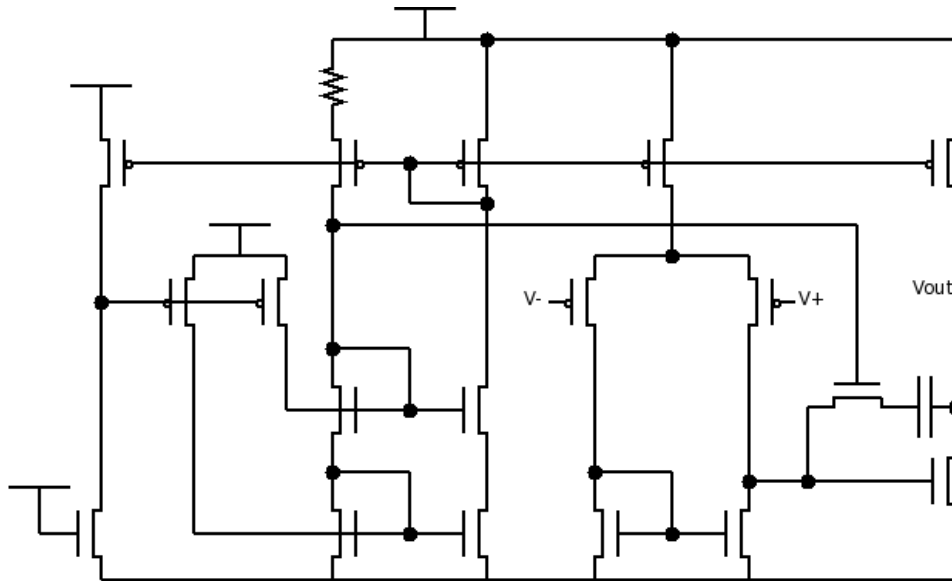


Figure 8.6: Schematics of the amplifier.

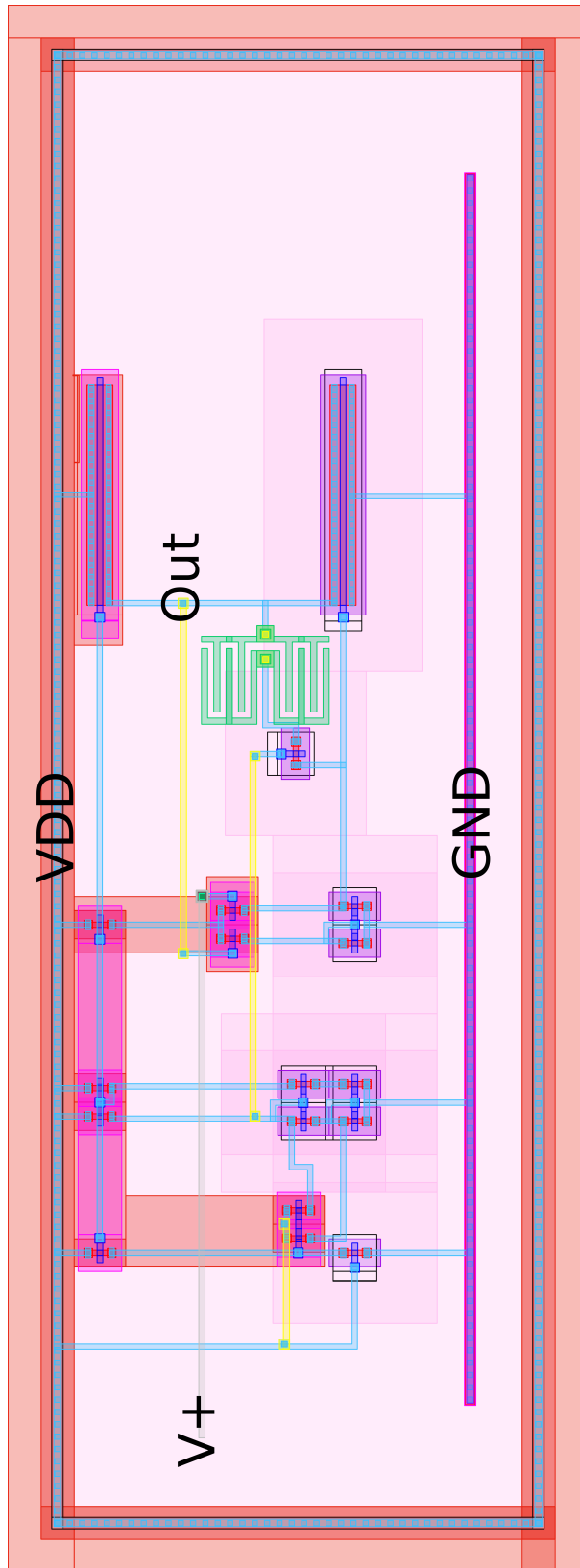


Figure 8.7: Layout of the amplifier.

### 8.1.5 Post-layout simulations

Simulation results for the layout can be seen in Figure 8.8. The clock frequency has been reduced for the layout simulation compared to the schematic simulation. The layout can operate at a frequency of up to  $10MHz$  unlike the schematic which works up to  $100MHz$ .

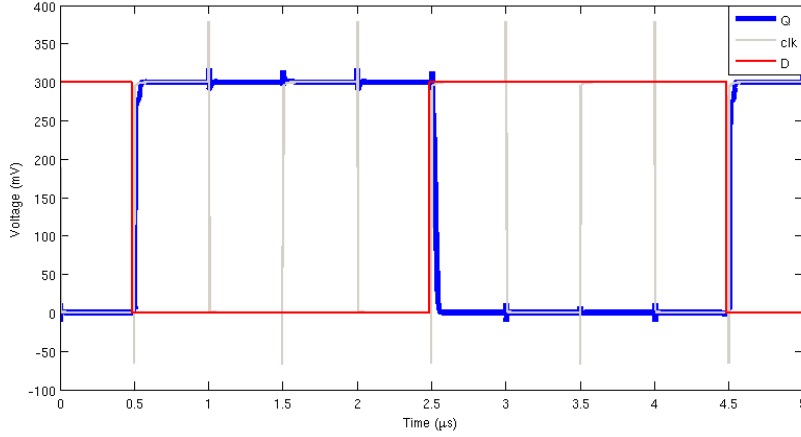


Figure 8.8: Simulation result showing the output signal for the layout at  $1MHz$  and  $300mV$ .

Table 8.1 shows a comparison of the schematic and layout for FFD. As we can see the layout has a much larger  $t_{setup}$ ,  $t_{c-q}$  and  $t_{d-q}$  compared to the schematic. When creating a layout there will always be parasitic elements which can exacerbate the final results. It is expected that circuits will perform worse in post-layout simulations than in schematical simulations.

Additionally the layout has not been optimized as well as the schematic since it was created in an early stage of the thesis. There were several considerations that were not taken into account, for instance the relationship between the evaluate transistors and the flip-flop inverters. There is also an imbalance between the rise and fall times. These are not however the main contributors to the increased delay and setup times compared to the schematic.

Flip-flop	$t_{setup}$	$t_{hold}$	$t_{c-q}$	$t_{d-q}$	Transistors
MSAFF	$0ns$	$2.6ns$	$10.0ns$	$10.0ns$	26
FFD schematic	$1.9ns$	$0.0ns$	$0.5ns$	$2.4ns$	16
FFD layout	$17ns$	$0.0ns$	$13ns$	$31ns$	16
schematic vs. layout	$9\times$	—	$26.0\times$	$13\times$	—

Table 8.1: Comparison between the schematic and layout for FFD at  $300mV$ .

A plot showing the output signal of the amplifier can be seen in Figure 8.9. One can see that after the signal is amplified the entire signal is shifted

about  $35mV$  above the amplifier input. However, in general, the shape remains the same. There are also ripples after the signal goes from low to high. These factors are not critical when doing measurements as we just want to see if the signal looks similar to the simulations. This would simply show that the ULV flip-flops works as intended. A close up of a rising edge can be seen in Figure 8.10. Here we can see that there is not much delay between the rising edge of the output signal and the amplified signal, but it takes about  $50ns$  for the signal to stabilize. On the falling edge there are no ripples.

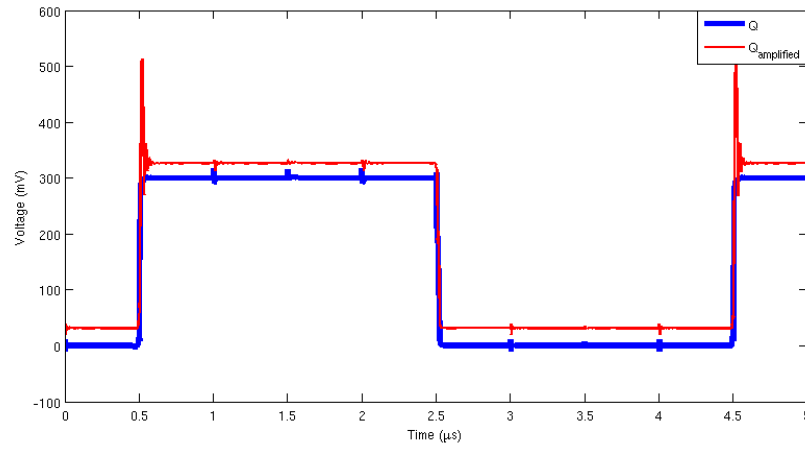


Figure 8.9: Simulation result showing the output signal and the amplified output signal for the layout at  $1MHz$  and  $300mV$ .

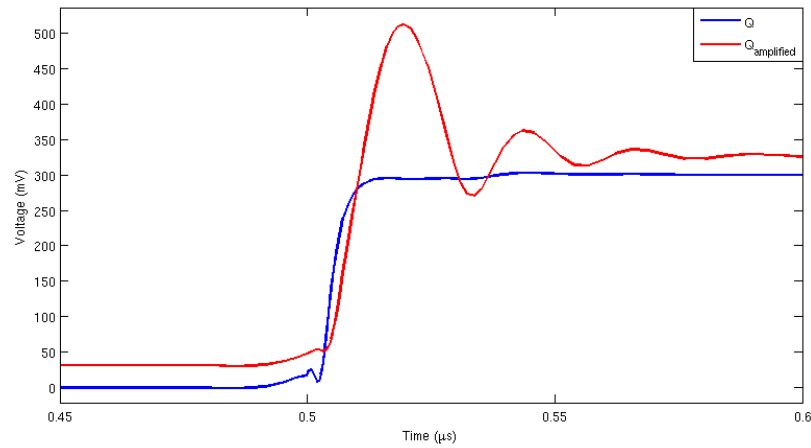


Figure 8.10: Close up simulation result showing the output signal and the amplified output signal for the layout at  $1MHz$  and  $300mV$ .

## 8.2 PCB and measurements

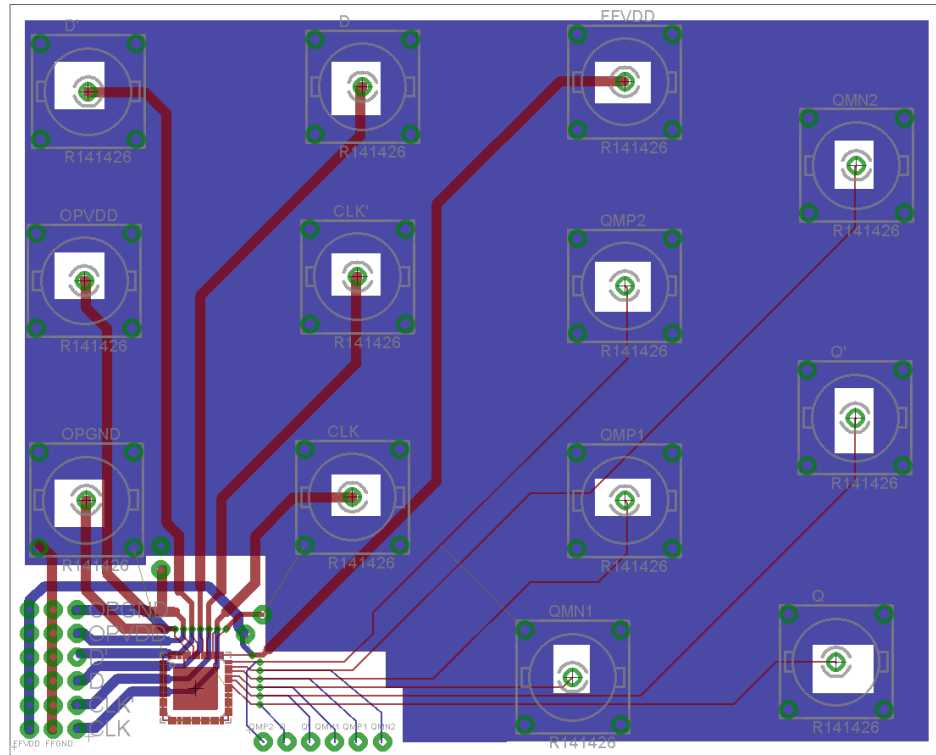


Figure 8.11: Layout of the PCB in EAGLE. Red is the top layer and blue is the bottom layer.

In order to do measurements on the chip a PCB had to be designed. The PCB was made in CadSoft EAGLE PCB Design. The layout of the PCB in EAGLE can be seen in Figure 8.11, and what the finished product looks like can be seen in Figure 8.12 and 8.13. The chip was soldered on to the PCB using a forced air convection reflow oven. The PCB is designed so that one can either use BNC connectors or the pins in the lower left corner of the PCB. It's good to have alternatives with different electrical properties in case one alternative doesn't work.

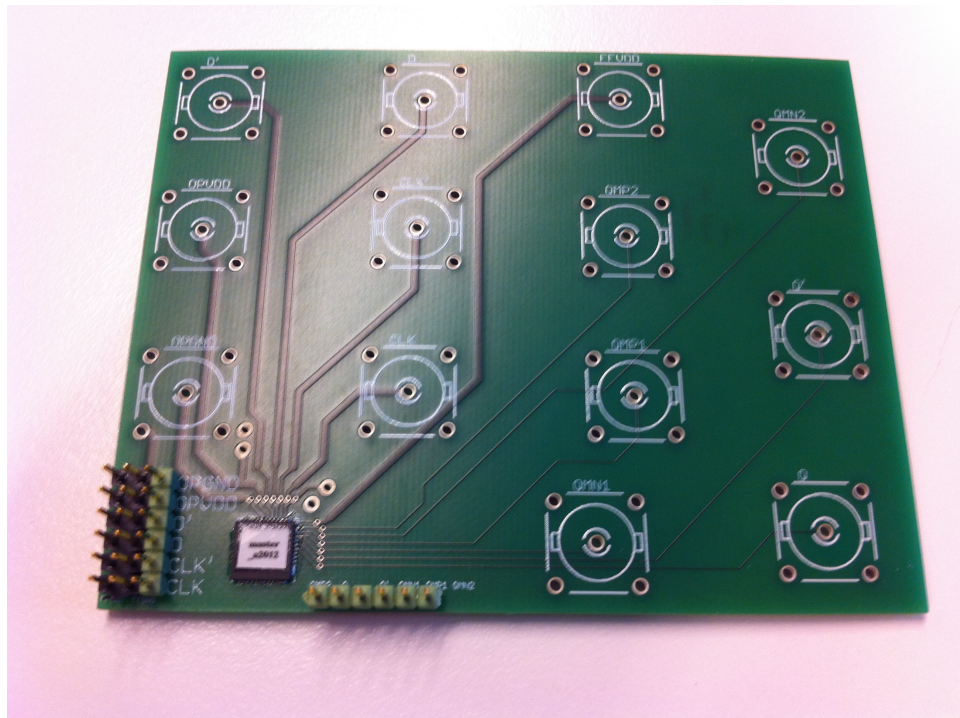


Figure 8.12: Picture of the PCB.

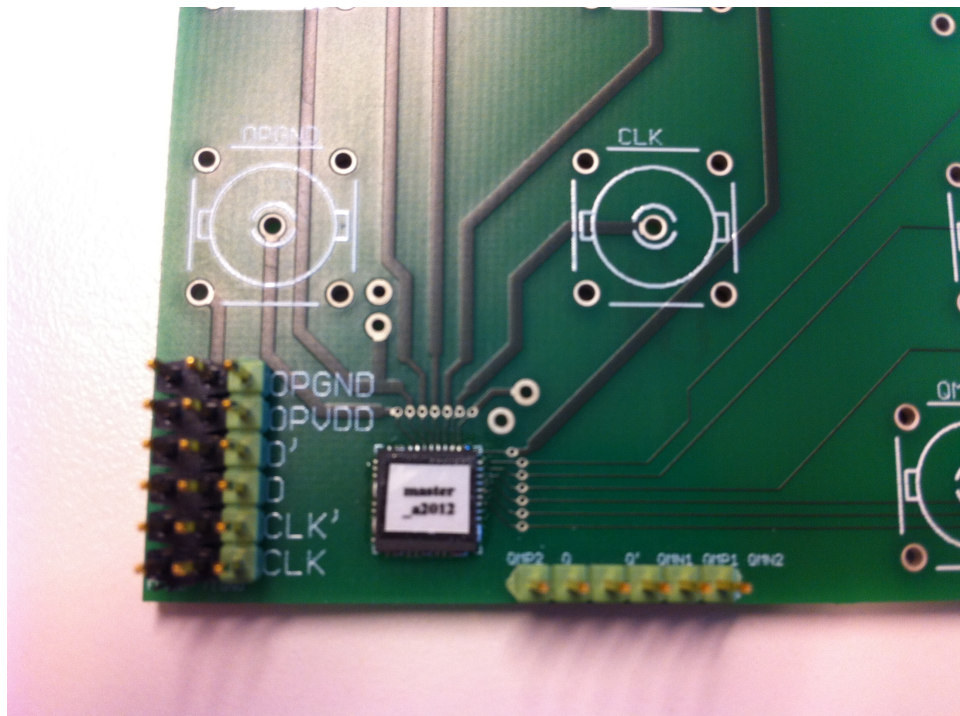


Figure 8.13: Close up of the chip on the PCB.



Figure 8.14 shows a picture of all the instruments used for measurements. The setup for measurements consists of 5 function generators, 2 power supplies and one oscilloscope. Since we want to synchronize all the 4 input signals we need a fifth function generator to be the control signal. We need 1 power supply for the flip-flop and 1 power supply for the amplifier since they use different  $V_{dd}$ .

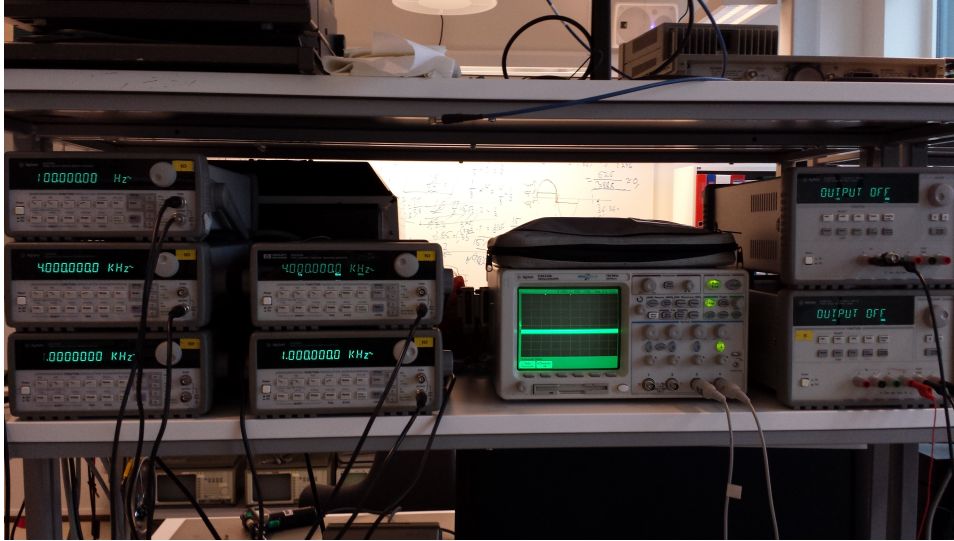


Figure 8.14: Picture of the setup used to do measurements.

Due to time limitaton and focus on measuring quality, we were not able to obtain relevant measurements. However, many details were examined during the process. Sources of error that has to be examined are for example the quality of the soldering, noise from the equipment, input load issues, correct equipment configurations etc. In addition there could potentially be problems with for example the amplifier design, PCB design or even issues with the layout design itself. All of this makes it very time consuming to acquire measurements on the chip, which is why no actual data can be presented in this thesis. With a larger time frame good measurements could have been obtained, and it is suggested that such measurements can be done for further works.





## Chapter 9

# Conclusion

This chapter summarizes some of the results and conclusions from previous chapters. Several similar ULV flip-flops have been presented in this thesis. Other flip-flop topologies have been presented for comparison.

Chapter 3 introduced ultra low voltage logic and gave a detailed explanation on how floating gates were used for ULV flip-flop designs. Floating gates were used to exploit the exponential property of the transistor current  $I_{ds}$  at subthreshold voltages by increasing the gate voltage by 50%.

Keeper transistors were added to the  $QM$  nodes on the flip-flops to empty the charge on the floating gate, and to turn off the evaluate transistors during the evaluate phase. This resulted in a much more stable and stronger output.

Flip-flop topologies were presented in Chapter 4. Among the other flip-flop topologies, MSAFF was the main focus for comparison with its non-existent setup time, reasonably low delay,  $t_{c-q} = 10ns$ , and very low hold time  $t_{hold} = 2.6ns$ . 4 different ULV flip-flops were presented, FFB, FFC, FFD and FFE. FFB and FFD proved to be 20 times faster than MSAFF. FFC and FFE were 14.3 and 16.7 times faster respectively.

Yield for all the flip-flops have been examined with regards to both frequency and supply voltage, ranging from  $10kHz$  to  $100MHz$  and  $100mV$  to  $400mV$ . FFC proved to be the best ULV flip-flop for all frequencies and supply voltages with a yield of 98% at  $100kHz$  and  $300mV$  with minimum input time. FFC was also able to get above 80% yield at  $100MHz$  and  $300mV$ , which is much higher than any other flip-flop.

The ULV flip-flops had very good EDP properties. Overall the flip-flops performed very equally except for FFC that showed very high power consumption compared to the other ULV flip-flops. The ULV flip-flops had an EDP of about  $1/8th$  of that of the comparison flip-flop. For static power the ULV flip-flops generally spent twice as much as that of the comparison flip-flop, however static power usually lies an order of magnitude below the dynamic power and will for most designs not be equally important.

When cascading several ULV flip-flops in a shift register several problems occurred. There was an issue with leakage from the clock signal into the input signal creating spikes and instability. Balancing of the

transistor strengths can somewhat deal with this issue, but with reduced  $c - q$  delay.

Chapter 5 covered the chip design and the layout was presented. Deep n-well implementation for floating bulk has been used by separating the p-substrates for each nMOS transistor with n-wells and deep n-wells. Also crosstalk between 2 metal wires, in metal layer 3, has been used to create capacitance between the clock signal and the evaluate transistors. The length of the 2 wires to get  $1fF$  was found to be  $17.6\mu m$  in metal layer 3 with minimum distance between the wires. The layout had an average  $t_{c-q}$  of  $13ns$ .

The work throughout this thesis has shown that the ULV flip-flops have different properties. This means that the circuit designer has to choose a flip-flop depending on the requirements of the circuit.

## 9.1 Further work

An even deeper analysis is wanted in order to use the ULV flip-flops in a VLSI. Future research topics includes:

- See how other flip-flops than FFB behaves in a shift register. Find solutions to various problems with registers.
- Further examine how the input time affects the yield.
- Optimize transistor strength sizing and bulk biasing for different supply voltages in order to investigate more accurately the yield and EDP with  $V_{dd}$  variations. In particular it should be possible to get 100% yield for FFC at  $400mV$ .
- Testing of integration with other ULV components.
- Further layout investigations on more ULV flip-flops, shift register and other options for bulk bias and capacitors. Furthermore the layout can be optimized better based on results in this thesis that were obtained at a later stage.
- Investigating the effectiveness of implementing ULV flip-flops in sleep mode circuits. This will take advantage of the low flip-flop delay that is present in these flip-flops.
- Analyze the chip more closely (the time scope of this thesis did not allow for any good measurements to be made).
- Investigate how the flip-flops would perform in other CMOS processes. Particularly processes that are much smaller ( $65nm/45nm/32nm/22nm$ ).

## Appendix A

# Comparison reference sheet

### FFB

#### Pros

- The fastest switching time(shared with FFE).
- Has very simple and straightforward input where the clock and *D* signals are not sent through the keeper or evaluate transistors. This gives low input load and also low load for the clock.

#### Cons

- The worst setup time.

### FFC

#### Pros

- Has the best yield.
- Has the shortest setup time.
- Has a lower dynamic power consumption than the other ULV flip-flops.
- Has a low spread on delay variation compared to the rest, and also has a higher chance of yielding flip-flops with a lower delay than any other flip-flop.

### **Cons**

- The slowest switching time of all ULV flip-flops.
- Has an added hold time of  $t_{hold} = 4.0ns$ .
- Does not benefit as much in EDP as the other ULV flip-flops.
- Has a very high static power consumption compared to the other flip-flops, particularly at lower frequencies.
- Has a higher input load making it harder to drive.

## **FFD**

### **Pros**

- The fastest switching time(shared with FFB).
- The second highest yield output, however FFC still wins by a very large margin.

### **Cons**

- Bad setup time, almost as long as that of FFB.

## **FFE**

### **Pros**

- Has a very short setup time.
- Keeper feedback system removes load from the clock and makes the keepers independent of clock. Overall more intuitive keeper system.
- At very low frequencies and very high input time this flip-flop still retains a respectable yield while the other 3 flip-flops falls down to 0% yield.

### **Cons**

- Has a high static power consumption at larger supply voltages.
- Has a high static power consumption at low operating frequencies.
- The lowest yield out of all the ULV flip-flops.

## Appendix B

# Comments on simulations

There are many ways to automatize the simulation process. For this thesis the process listed below was used. This could be of use for the reader who wishes to automatize their own simulation process.

1. The simulation testbench schematic was set up in Cadence.
2. An Ocean script was generated in Cadence Analog Environment (Session→Save Script) in order to get all the relevant pathing information for circuit components. The script was then modified to suit the needs of the desired simulation.
3. The schematics must be netlisted at least once. This can be done by simply running a simulation of it in Analog Environment, or by going to “Simulation→Netlist→Create”.
4. The Ocean script can be run from the terminal using “source .bash\_tsmc90nmlp; ocean -restore file.ocn”. This in turn can be run directly from MATLAB using the system and sprintf functions: “system(sprintf('string'))”.
5. Sometimes it is useful to be able to edit variables in the Ocean script during a MATLAB simulation. This can be done by MATLAB using sed commands in system:

```
system(sprintf('sed -r %s /("%s"[_\\t]*)[0-9]+/\\n1%d/ ' %s > _temp.ocn; mv _temp.ocn %s ', param, value, ocn, ocn));
```

6. The output data from Ocean was scripted to be stored in a “data.out” file. This could be accessed from matlab using “importdata()”.

It should be noted that pathing in Ocean happens from the tsmc90nmlp folder, so if the script is located elsewhere make sure the pathing of the script is modified correctly.

### **Problem with Cadence and MATLAB for power simulations when sweeping the supply voltage**

Using a low frequency for power simulations creates a problem related to limitations in Cadence and MATLAB. The time step between each point in a simulation determines the resolution of the simulation. In order to correctly measure the delay at high supply voltages a small time step is needed ( $\sim 10ps$ ) or else the delay will be unreliable. However at the same time, because the operating frequency is so low, the total simulation time has to be very high ( $\sim 300\mu s$ ). This creates an extremely high resolution for simulation ( $\sim 30'000'000$  points) which presents a huge problem when importing the simulation results from Cadence to matlab. Generally things starts to slow down if the simulation exceeds 10'000 points, so with the amount of points retained in these simulations the export of data from Cadence to matlab never occurs within a reasonable amount of time.

## Appendix C

# Ocean script for simulations

### C.1 Yield simulations

Ocean script for Monte Carlo simulations:

#### **MonteCarlo.ocn**

;Replace ~ with the correct path

VDD = 400m

clk = 10000.00n

ocnWaveformTool( 'wavescan' )

simulator( 'spectre' )

design( "~/spectre/schematic/netlist/netlist" )

resultsDir( "~/spectre/schematic" )

modelFile(

    ;SCHEMATIC AND OTHER SIMULATION FILE LOCATIONS HERE

)

analysis( 'tran ?stop clk\*10' )

desVar( "lengthK" 100n )

desVar( "vdd" VDD )

desVar( "setup" 1.2n )

desVar( "Rp" 900n )

desVar( "Rn" 540n )

desVar( "Kp" 120n )

desVar( "Kn" 120n )

desVar( "Ep" 120n )

desVar( "En" 120n )

desVar( "clkperiod" clk )

desVar( "C" 1f )

out = outfile( "./ULVflipflopResults.out" "w" )

temp( 27 )

monteCarlo( ?numIters "100" ?startIter "1"

    ?analysisVariation 'processAndMismatch ?sweptParam "  
    None"

    ?sweptParamVals "27" ?saveData t

    ?nomRun "yes" ?append nil

```

        ?saveProcessParams t
    )
monteRun()
selectResult( 'tran )
outputs()
ocnPrint( ?output "./ULVflipflopResults.out" v("Q") ?
    precision 16 ?numberNotation 'scientific ?from 0 ?to clk
    *10 ?step clk/100 )
close(out)
exit

```

---

## C.2 Power simulations

Ocean script for simulation of static power:

### **edp.ocn**

;Replace ~ with the correct path

```

ocnWaveformTool( 'wavescan )
simulator( 'spectre )
design( "~/spectre/schematic/netlist/netlist" )
resultsDir( "~/spectre/schematic" )
modelFile(
    ;SCHEMATIC AND OTHER SIMULATION FILE LOCATIONS HERE
)
analysis('tran ?stop "150u" )
desVar( "vdd" 300m )
desVar( "setup_time" 10u )
desVar( "period" 80u )
desVar( "fallrise" 1f )
desVar( "clk" 20u )
desVar( "c" 1f )
saveOption( 'pwr "all" )
temp( 27 )
out = outfile("~/edp.out" "w")
paramAnalysis("vdd" ?values '( 0.0500000000000000
    0.0600000000000000 0.0700000000000000
    0.0800000000000000 0.0900000000000000
    0.1000000000000000 0.1100000000000000
    0.1200000000000000 0.1300000000000000
    0.1400000000000000 0.1500000000000000
    0.1600000000000000 0.1700000000000000
    0.1800000000000000 0.1900000000000000
    0.2000000000000000 0.2100000000000000
    0.2200000000000000 0.2300000000000000
    0.2400000000000000 0.2500000000000000
    0.2600000000000000 0.2700000000000000
    0.2800000000000000 0.2900000000000000

```



```

0.3000000000000000    0.3100000000000000
0.3200000000000000    0.3300000000000000
0.3400000000000000    0.3500000000000000
0.3600000000000000    0.3700000000000000
0.3800000000000000    0.3900000000000000
0.4000000000000000
)
)
paramRun()
selectResult( 'tran )
outputs()
pow = getData("I92:pwr" ?result "tran-tran")    ;I92 is the
        name of the Flip-flop
ocnPrint( ?output "~/edp.out" v("Q") pow ?precision 16 ?
        numberNotation 'scientific ?from 90u ?to 93u ?step 0.35n
        )
close(out)
exit

```

---

Ocean script for simulation of dynamic power, delay and EDP:

### **edp2.ocn**

;Replace ~ with the correct path

```

ocnWaveformTool( 'wavescan )
simulator( 'spectre )
design( "~/spectre/schematic/netlist/netlist")
resultsDir( "~/spectre/schematic" )
modelFile(
        ;SCHEMATIC AND OTHER SIMULATION FILE LOCATIONS HERE
)
analysis('tran ?stop "150u" )
desVar( "vdd" 300m )
desVar( "setup_time" 10u )
desVar( "period" 80u )
desVar( "fallrise" 1f )
desVar( "clk" 20u )
desVar( "c" 1f )
saveOption( 'pwr "all" )
temp( 27 )
out = outfile("~/edp.out" "w")
paramAnalysis("vdd" ?values '( 0.0500000000000000
0.0600000000000000    0.0700000000000000
0.0800000000000000    0.0900000000000000
0.1000000000000000    0.1100000000000000
0.1200000000000000    0.1300000000000000
0.1400000000000000    0.1500000000000000
0.1600000000000000    0.1700000000000000
0.1800000000000000    0.1900000000000000

```

```

0.2000000000000000    0.2100000000000000
0.2200000000000000    0.2300000000000000
0.2400000000000000    0.2500000000000000
0.2600000000000000    0.2700000000000000
0.2800000000000000    0.2900000000000000
0.3000000000000000    0.3100000000000000
0.3200000000000000    0.3300000000000000
0.3400000000000000    0.3500000000000000
0.3600000000000000    0.3700000000000000
0.3800000000000000    0.3900000000000000
0.4000000000000000
)
)
paramRun()
selectResult( 'tran )
outputs()
pow = getData("I92:pwr" ?result "tran-tran")    ;I92 is the
        name of the Flip-flop
ocnPrint( ?output "~/edp.out" v("Q") pow ?precision 16 ?
        numberNotation 'scientific ?from 100u ?to 120u ?step 2.1
        n )
close(out)
exit

```

---

Ocean script for simulation of static power for frequency sweep:

### **freq.ocn**

;Replace ~ with the correct path

```

freq = 50000000.000000
temporary = freq/10000
T = 100u/temporary

ocnWaveformTool( 'wavescan )
simulator( 'spectre )
design( "~/spectre/schematic/netlist/netlist")
resultsDir( "~/spectre/schematic" )
modelFile(
        ;SCHEMATIC AND OTHER SIMULATION FILE LOCATIONS HERE
)
analysis('tran ?stop 7.5*T )
desVar( "vdd" 300m )
desVar( "setup_time" 10n )
desVar( "period" 4*T )
desVar( "fallrise" 1f )
desVar( "clk" T )
desVar( "c" 1f )
saveOption( 'pwr "all" )
temp( 27 )

```

```

out = outfile("~/edp.out" "w")
run()
selectResult( 'tran )
outputs()
pow = getData("I92:pwr" ?result "tran-tran")
ocnPrint( ?output "~/edp.out" v("Q") pow ?precision 16 ?
    numberNotation 'scientific ?from 0 ?to 7.5*T ?step 7.5*T
    /9000 )
close(out)
exit

```

---

Ocean script for simulation of dynamic power, delay and EDP for frequency sweep:

### **freq2.ocn**

```

;Replace ~ with the correct path

freq = 500000000.000000
temporary = freq/10000
T = 100u/temporary

ocnWaveformTool( 'wavescan )
simulator( 'spectre )
design( "~/spectre/schematic/netlist/netlist")
resultsDir( "~/spectre/schematic" )
modelFile(
    ;SCHEMATIC AND OTHER SIMULATION FILE LOCATIONS HERE
)
analysis('tran ?stop 7.5*T )
desVar( "vdd" 300m )
desVar( "setup_time" 5n )
desVar( "period" 4*T )
desVar( "fallrise" 1f )
desVar( "clk" T )
desVar( "c" 1f )
saveOption( 'pwr "all" )
temp( 27 )
out = outfile("~/edp.out" "w")
run()
selectResult( 'tran )
outputs()
pow = getData("I92:pwr" ?result "tran-tran")
ocnPrint( ?output "~/edp.out" v("Q") pow ?precision 16 ?
    numberNotation 'scientific ?from 2*T+5n ?to 2*T+25n ?
    step 0.0025n )
close(out)
exit

```

---



## Appendix D

# MATLAB code for simulations

### D.1 Yield simulations

This program runs the Ocean script:

```
RunOceanScript.m  
function[TempStruct] = RunOceanScript(oceanScript)  
  
system(sprintf('source_.bash_tsmc90nmlp;_ocean_-restore_%s'  
    , oceanScript));  
t = importdata('ULVflipflopResults.out', '_', 4);  
TempStruct = t.data;  
end
```

---

This program sweeps the frequency by calling RunOceanScript.m in a loop for the different frequencies:

```
FreqSweep.m  
function[yield] = FreqSweep(fromFreq, toFreq, step)  
  
freq = logspace(fromFreq, toFreq, step);  
vdd = 300;  
  
for i=1:length(freq)  
    system(sprintf('sed_-i_' '1c\\VDD_=%f' ' _MonteCarlo.  
        ocn', vdd));  
    system(sprintf('sed_-i_' '2c\\clk_=%f' ' _MonteCarlo.  
        ocn', 1e9/freq(i)));  
    FF = RunOceanScript('MonteCarlo.ocn');  
    yield(i) = FindYield(FF, 1e9/freq(i), vdd);  
end  
  
semilogx(freq, yield)
```

This program sweeps the supply voltage by calling RunOceanScript.m in a loop for the different voltages:

#### **VddSweep.m**

```
function[yield] = VddSweep(fromVdd, toVdd, step)

Vdd = linspace(fromVdd, toVdd, step);
clkperiod = 10e3;

for i=1:length(Vdd)
    system(sprintf('sed_-i_' '1c\\VDD=_%.0fm' ' _MonteCarlo.
        ocn',Vdd(i)));
    system(sprintf('sed_-i_' '2c\\clk=_%.2fn' ' _MonteCarlo.
        ocn',clkperiod));
    FF = RunOceanScript('MonteCarlo.ocn');
    yield(i) = FindYield(FF, clkperiod, Vdd(i));
end

plot(Vdd,yield)
```

---

This program receives arguments, including the results, from FreqSweep.m and VddSweep.m and calculates the yield:

#### **FindYield.m**

```
function[yield] = FindYield(FF, clkperiod, vdd)

f = 1;
failed = [];
vdd = vdd/1000;
clkperiod = clkperiod*1e-9;
st = 0.95e-6;

% First segment
for j=1:size(FF,1)
    if st < FF(j,1)
        start1 = j;
        break;
    end
end
for p=1:size(FF,1)
    if clkperiod*2 <= FF(p,1)
        stop1 = p;
        break;
    end
end

% Second segment
for h=1:size(FF,1)
    if clkperiod*2 + st < FF(h,1)
```

```

        start2 = h;
        break;
    end
end
for u=1:size(FF,1)
    if clkperiod*4 <= FF(u,1)
        stop2 = u;
        break;
    end
end

% Third segment
for v=1:size(FF,1)
    if clkperiod*4 + st < FF(v,1)
        start3 = v;
        break;
    end
end
for g=1:size(FF,1)
    if clkperiod*6 <= FF(g,1)
        stop3 = g-1;
        break;
    end
end

for i=2:size(FF,2)
    loop1 = 0;
    loop2 = 0;
    for k=start1:stop1
        if vdd - vdd*0.2 < FF(k,i) && FF(k,i) < vdd + vdd
            *0.2
        else
            failed(f) = i-1;
            f = f+1;
            loop1 = 1;
            break;
        end
    end
    if loop1 == 0
        for y=start2:stop2
            if -vdd*0.2 < FF(y,i) && FF(y,i) < vdd*0.2
            else
                failed(f) = i-1;
                f = f+1;
                loop2 = 1;
                break;
            end
        end
    end
    if loop1 == 0 && loop2 == 0
        for k=start3:stop3

```

```

        if vdd - vdd*0.2 < FF(k,i) && FF(k,i) < vdd +
            vdd*0.2
        else
            failed(f) = i-1;
            f = f+1;
            break;
        end
    end
end
end
end
yield = 100-size(failed,2);

```

---

This program calculates the delay for all the Monte Carlo simulations:

```

CtoQ.m
function [D, error] = CtoQ(Qin, clk, failed, expCtoQ)

vdd = 0.3;
vddHalf = vdd/2.;
error = length(failed);
p = 1;
u = 1;

for i=1:length(Qin)
    if Qin(i,1) >= clk(1)
        if p
            start1 = i;
            p = 0;
        end
    end
end

for i=1:length(Qin)
    if Qin(i,1) >= clk(2)
        if u
            start2 = i;
            u = 0;
        end
    end
end

for i=2:size(Qin,2)
    working = 1;
    for k=1:length(failed)
        if i-1 == failed(k)
            working = 0;
        end
    end
    if working

```



```

        for j=start1:length(Qin)
            if Qin(j,i) <= vddHalf
                D1 = Qin(j,1)-clk(1);
                break
            end
        end
        for j=start2:length(Qin)
            if Qin(j,i) >= vddHalf
                D2 = Qin(j,1)-clk(2);
                break
            end
        end
        D(i-1) = (D1+D2)/2.;
    end
end

D=1e9*D(D~=0);          % Remove zeros

[n,x] = hist(D,10);
bar(x,100*n/(100-length(failed)),'hist');

hold on

m1 = 100*max(n)/(100-length(failed));
m = 10*ceil(m1/10);

h = findobj(gca, 'Type', 'patch');
set(h, 'FaceColor', [0.86 0.86 0.86])

% Plots a vertical line of the expected D value
plot([1e9*expCtoQ, 1e9*expCtoQ], [0, m], '--', ...
    'Color', [0, 0.5, 0], 'LineWidth', 3)

xlabel 't_{c-q}(ns)'
ylabel 'Percentage_of_flip-flops_that_falls_within_a_given_delay'

```

---

## D.2 Power simulations

Basic simulation file for running the Ocean scripts:

### **simulate.m**

```
function[data] = simulate(ocn)
system(sprintf('cd_...;_cd_...;_source_.bash_tsmc90nmlp;_
    ocean_-restore_%s',ocn));
temp = importdata('./edp.out', '_', 4); %Or name of output
    file generated by Ocean
data = temp.data;
```

---

Starts up simulations for frequency sweep of a flip-flop and stores the data in arrays:

### **simFreq.m**

```
n = 120; %Number of simulations to run.
f = logspace(log10(50e3),log10(50e6),n); %Frequency sweep
    array.
t = zeros(9000,n);
pRaw = t;
tS = zeros(8000,n);
Q = tS;
PS = tS;

%Simulations and calculations for dynamic(d) power.
for i=1:n %Run simulations and sweep frequency.
    setFreq(f(i));
    temp = simulate('freq.ocn'); %Makes a full simulation
        over several periods.
    t(:,i) = temp(:,1); %Time data.
    pRaw(:,i) = temp(:,3); %Wave data for the power.
    temp2 = simulate('freq2.ocn'); %Simulates high
        resolution for the moment of switching.
    tS(:,i) = temp2(:,1);
    Q(:,i) = temp2(:,2); %Wave data for the Q signal
    PS(:,i) = temp2(:,3); %Wave data for the power during
        switching.
end

dD = linspace(0,0,n);
dP = dD;
dEDP = dD;
dPDP = dD;
sP = dD;
P = dD;

for i=1:n
    [C,t2] = min(abs(t(:,i)-(2.5/f(i)))));
    [C,t3] = min(abs(t(:,i)-(3.5/f(i)))));
```

```

    for j=1:length(tS(:,i));
        if Q(j,i) <= 0.15
            dD(i) = (tS(j,i) - tS(1,i)); %finds the delay
            break;
        end
    end
end

for j=1:length(tS(:,i)); %checks total duration of
    active switching period on the Q signal to 95% of
    vdd.
    if Q(j,i) <= 0.3*0.05
        dP(i) = mean(PS(1:j,i)); %finds avarage dynamic
            power dissipation for the flip-flop.
        break;
    end
end

sP(i) = mean(pRaw(t2:t3,i)); %finds avarage static
    power dissipation for the flip-flop.

P(i) = mean(pRaw(:,i));

dPDP(i) = dP(i)*dD(i);
dEDP(i) = dPDP(i)*dD(i);
end

```

---

Starts up simulations for  $V_{dd}$  sweep of a flip-flop and stores the data in arrays:

#### **simVdd.m**

```

vdd = linspace(50,400,36); %Region for the power supply
    sweep.

%Simulations and calculations for dynamic(d) power.
data = simulate('edp.ocn');
dD = linspace(0,0,36);
dP = dD;
dEDP = dD;
dPDP = dD;

t = data(:,1); %time data

for i=1:36
    Q = data(:,i*2); %wave data for the Q signal.
    [C,t1] = min(abs(t-90e-6)); %returns the index of t=90
        us and puts it in t1

    for j=t1:length(t); %checks c-q delay at 50% vdd
        if Q(j) >= 0.001*vdd(i)/2.0

```

```

        dD(i) = t(j) - 90e-6;
        break;
    end
end
for j=t1:length(t); %checks total duration of active
switching period on the Q signal to 95% of vdd.
    if Q(j) >= 0.001*vdd(i)*0.95
        dP(i) = mean(data(1:j,1+2*i)); %finds avarage
            dynamic power dissipation for the flip-flop.
        break;
    end
end
end

dPDP(i) = dP(i)*dD(i);
dEDP(i) = dPDP(i)*dD(i);
end

%Simulations and calculations for static(s) power.
data = simulate('edp2.ocn');
sP = linspace(0,0,36);

t = data(:,1); %time data

for i=1:36
    sP(i) = mean(data(:,1+2*i)); %finds avarage static
        power dissipation for the flip-flop.
end

```

---

## Appendix E

### Additional layout close up

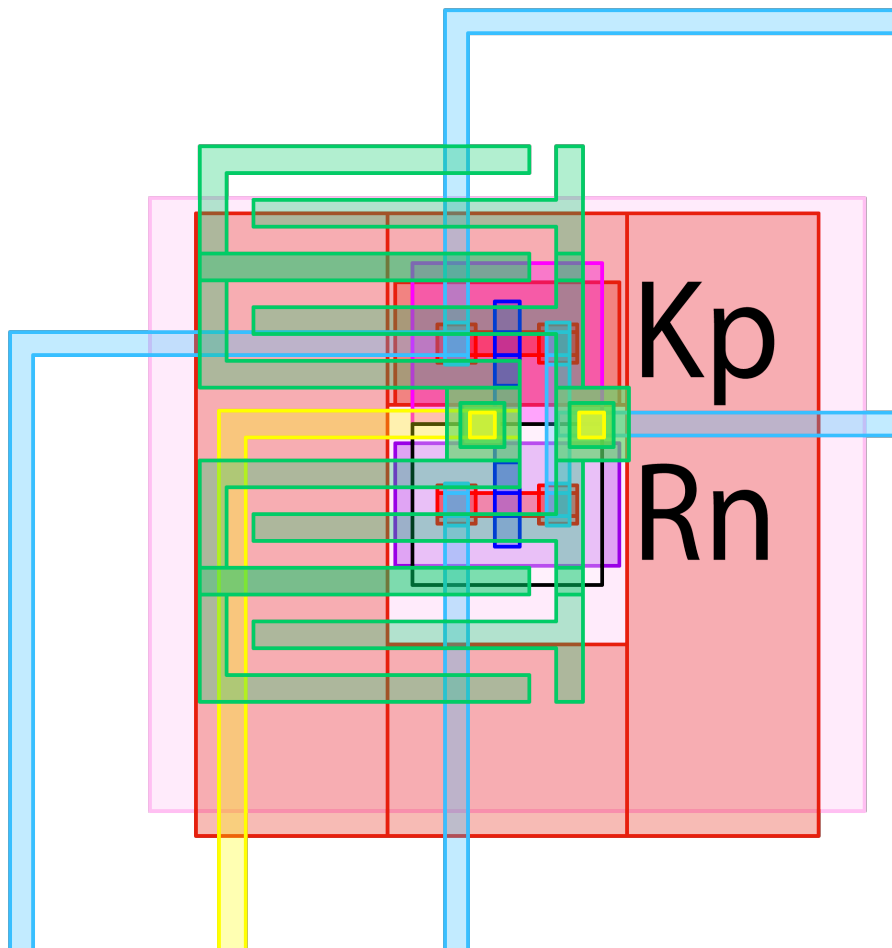


Figure E.1: Close up of the upper left corner of the flip-flop layout. An nMOS recharge transistor and a pMOS Keeper transistor is shown.

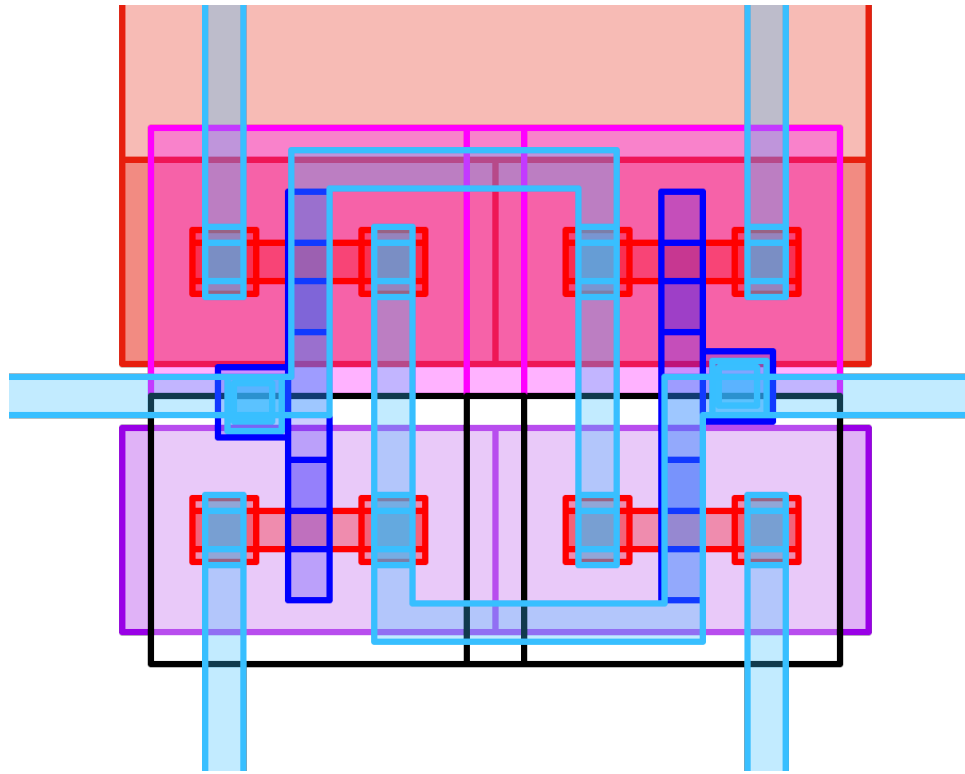


Figure E.2: Close up of the flip-flop inverters at the center of the flip-flop layout.

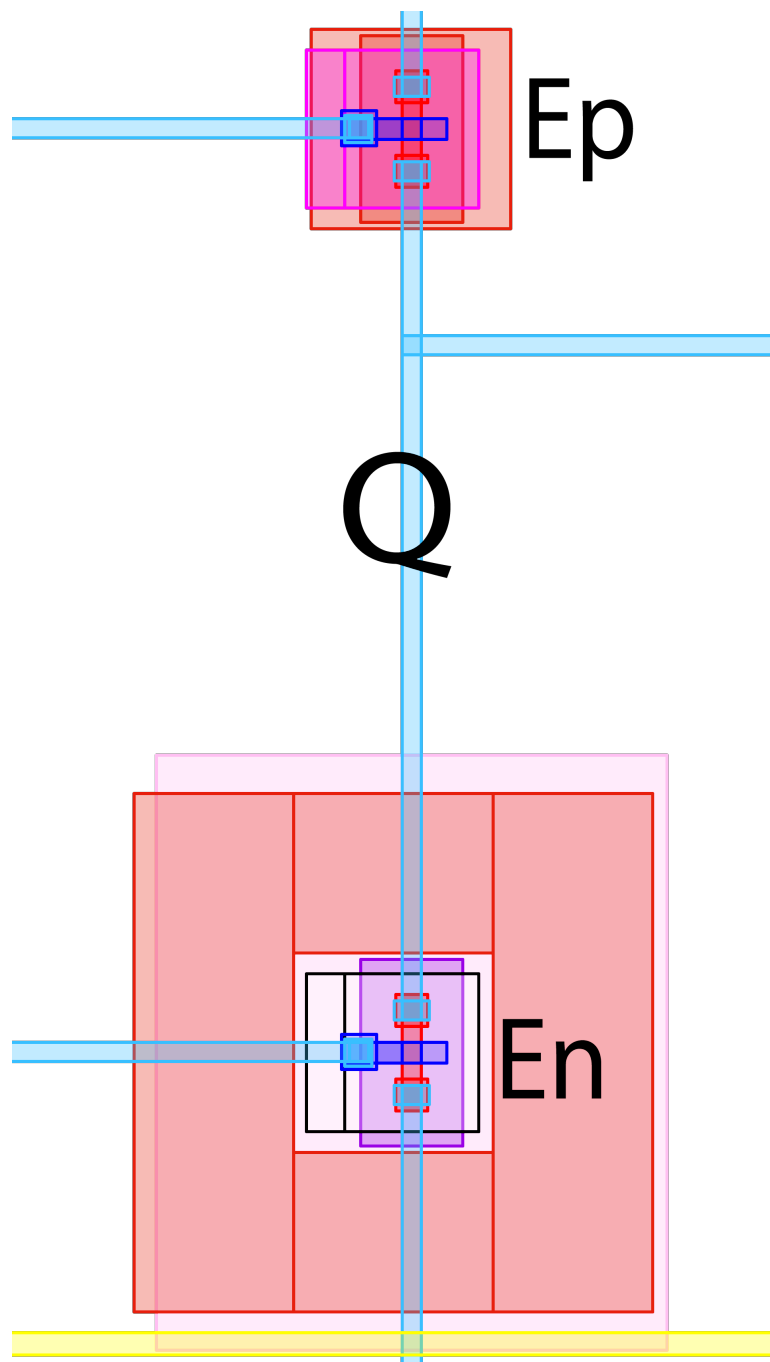


Figure E.3: Close up of the evaluate transistors along with the output signal of the flip-flop layout.

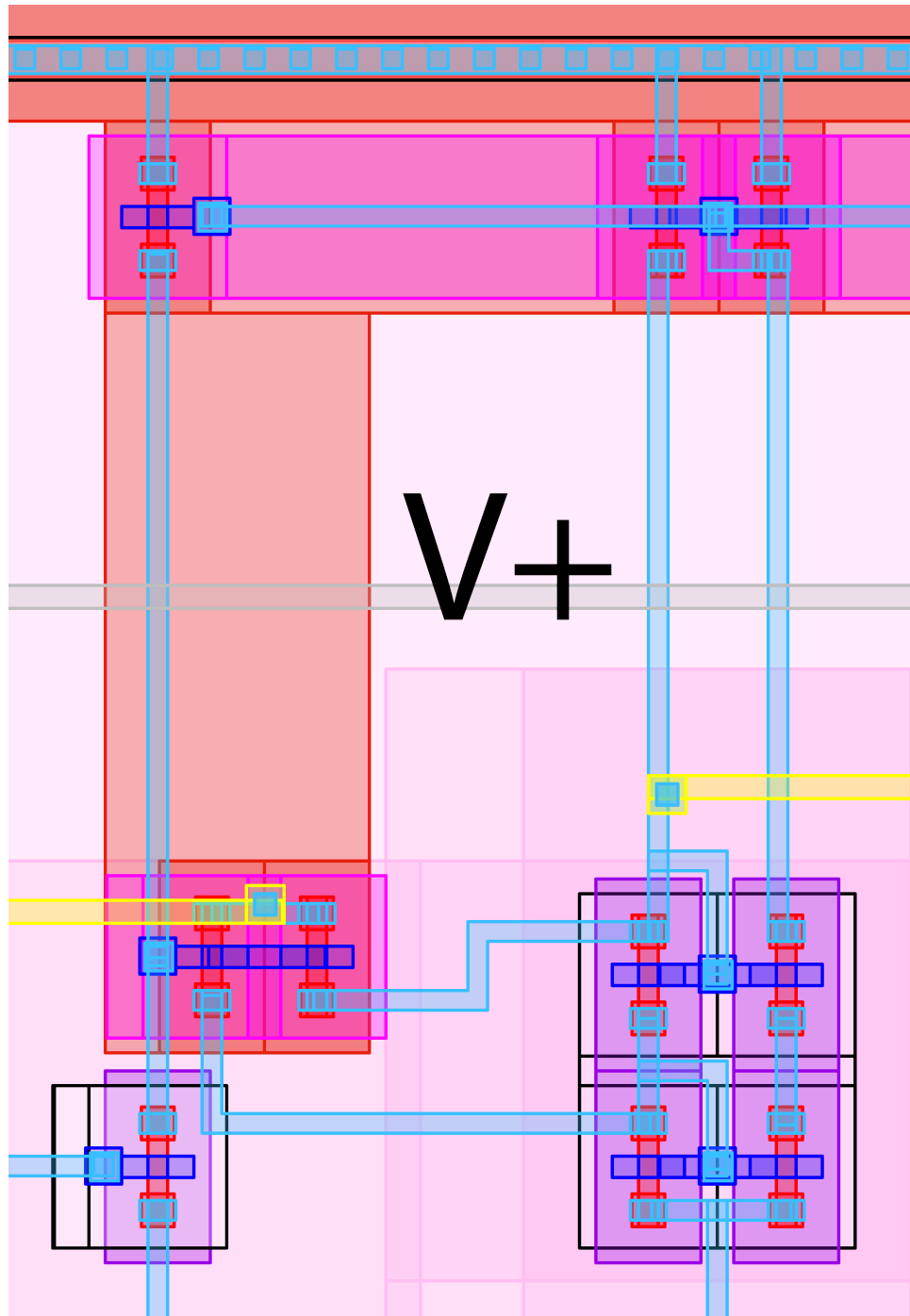


Figure E.4: Close up of the amplifier layout, left side.



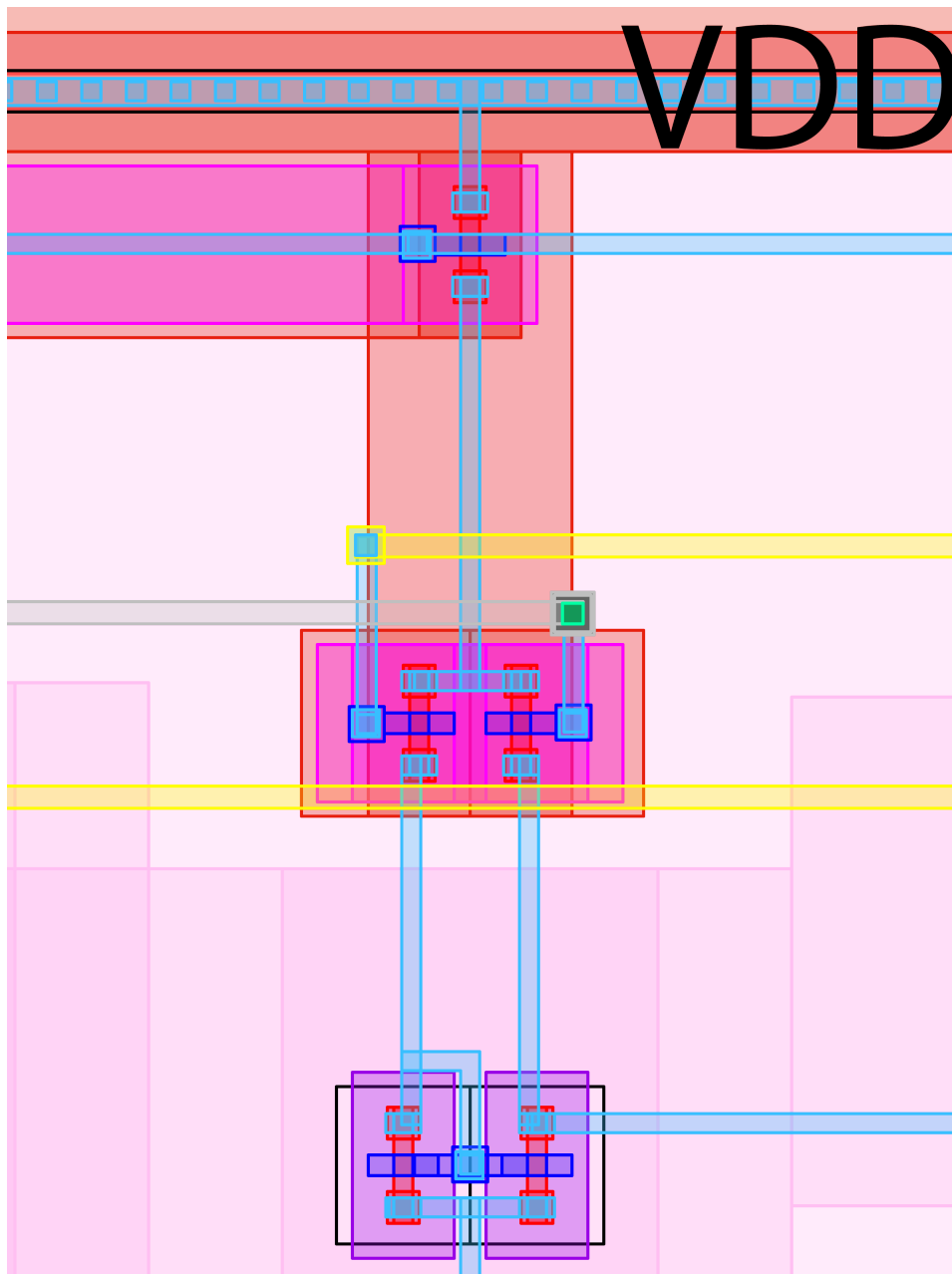


Figure E.5: Close up of the amplifier layout, center.

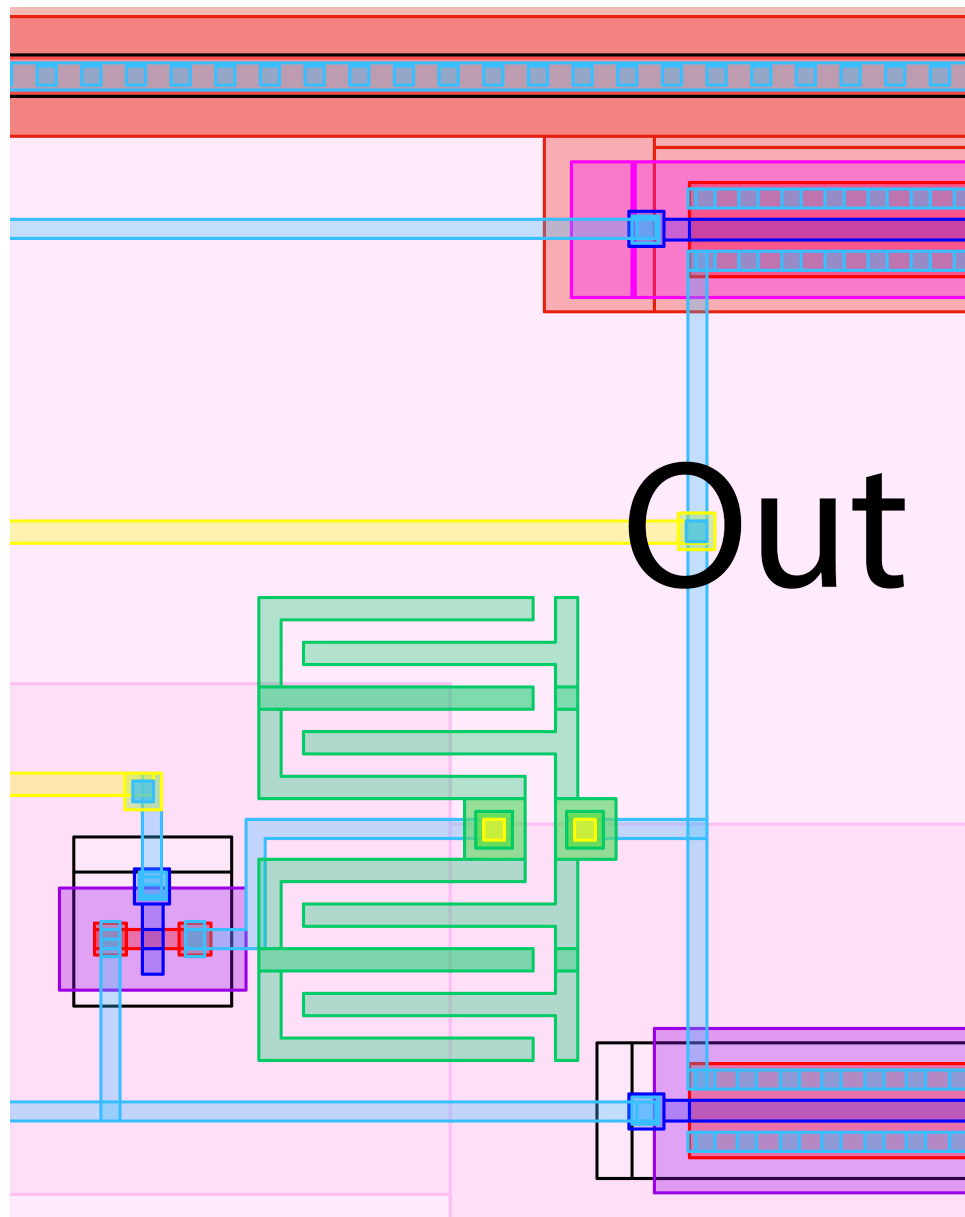


Figure E.6: Close up of the amplifier layout, right side.

# Bibliography

- [1] Yoonmyung Lee, G. Chen, S. Hanson, D Sylvester, and D Blaauw. Ultra-low power circuit techniques for a new class of sub-mm<sup>3</sup> sensor nodes. In *Custom Integrated Circuits Conference (CICC), 2010 IEEE*, pages 1–8, 2010.
- [2] A.P. Chandrakasan, S. Sheng, and R.W. Brodersen. Low-power cmos digital design. *Solid-State Circuits, IEEE Journal of*, 27(4):473 –484, Apr. 1992.
- [3] N. Verma, J. Kwong, and A.P. Chandrakasan. Nanometer mosfet variation in minimum energy subthreshold circuits. *Electron Devices, IEEE Transactions on*, 55(1):163–174, 2008.
- [4] S. Mutoh, T. Douseki, Y. Matsuya, T. Aoki, S. Shigematsu, and J. Yamada. 1-v power supply high-speed digital circuit technology with multithreshold-voltage cmos. *Solid-State Circuits, IEEE Journal of*, 30(8):847–854, 1995.
- [5] Kimiyoshi Usami and Mark Horowitz. Clustered voltage scaling technique for low-power design. In *Proceedings of the 1995 international symposium on Low power design, ISLPED '95*, pages 3–8, New York, NY, USA, 1995. ACM.
- [6] D. Bhargavaram and M.G.K. Pillai. Low power dual edge triggered flip-flop. In *Advances in Engineering, Science and Management (ICAESM), 2012 International Conference on*, pages 63–67, 2012.
- [7] S. Naik and R. Chandel. Design of a low power flip-flop using cmos deep sub micron technology. In *Recent Trends in Information, Telecommunication and Computing (ITC), 2010 International Conference on*, pages 253–256, 2010.
- [8] Sudeep Balan and Sanil K Daniel. Dual-edge triggered sense-amplifier flip-flop for low power systems. In *Green Technologies (ICGT), 2012 International Conference on*, pages 135–142, 2012.
- [9] Xue-Xiang Wu and Ji-Zhong Shen. Low-power explicit-pulsed triggered flip-flop with robust output. *Electronics Letters*, 48(24):1523–1525, 2012.

- [10] Y. Berg. A novel high speed differential ultra low-voltage cmos flip-flop for high speed applications. In *The Fifth International Conference on Advances in Circuits, Electronics and Micro-electronics*, pages 11–16, Aug. 2012.
- [11] Y. Berg, O. Mirmotahari, and S. Aunet. Clocked semi-floating-gate ultra low-voltage current mirror. In *Electronics, Circuits and Systems, 2008. ICECS 2008. 15th IEEE International Conference on*, pages 1038–1041, 2008.
- [12] Y. Berg, T.-S. Lande, and O. Naess. Programming floating-gate circuits with uv-activated conductances. *Circuits and Systems II: Analog and Digital Signal Processing, IEEE Transactions on*, 48(1):12–19, 2001.
- [13] M. Azadmehr and Y. Berg. An ultra-low voltage pseudo-floating gate amplifier. In *Faible Tension Faible Consommation (FTFC), 2012 IEEE*, pages 1–4, 2012.
- [14] Y. Berg and M. Azadmehr. Novel ultra low-voltage and high-speed cmos pass transistor logic. In *Faible Tension Faible Consommation (FTFC), 2012 IEEE*, pages 1–4, 2012.
- [15] Y. Berg and O. Mirmotahari. Ultra low-voltage and high speed dynamic and static cmos precharge logic. In *Faible Tension Faible Consommation (FTFC), 2012 IEEE*, pages 1–4, 2012.
- [16] Y. Berg. Ultra low-voltage and high-speed cmos full adder using floating-gates and multiple-valued logic. In *Multiple-Valued Logic (ISMVL), 2011 41st IEEE International Symposium on*, pages 259–262, 2011.
- [17] M. Azadmehr and Y. Berg. A band pass auto-zeroing floating-gate amplifier. In *Faible Tension Faible Consommation (FTFC), 2011*, pages 83–86, 2011.
- [18] G.E. Moore. Cramming more components onto integrated circuits. *Proceedings of the IEEE*, 86(1):82–85, 1998.
- [19] M. Alioto. Ultra-low power vlsi circuit design demystified and explained: A tutorial. *Circuits and Systems I: Regular Papers, IEEE Transactions on*, 59(1):3–29, Jan. 2012.
- [20] Wikipedia, “transistor count and moore’s law”. [http://en.wikipedia.org/wiki/File:Transistor\\_Count\\_and\\_Moore's\\_Law\\_-\\_2011.svg](http://en.wikipedia.org/wiki/File:Transistor_Count_and_Moore's_Law_-_2011.svg). Accessed: July-2013.
- [21] A. Bryant, J. Brown, P. Cottrell, M. Ketchen, J. Ellis-Monaghan, and E.J. Nowak. Low-power cmos at  $v_{dd} = 4kT/q$ . In *Device Research Conference, 2001*, pages 22–23, 2001.

- [22] M. Steyaert, V. Peluso, J. Bastos, P. Kinget, and W. Sansen. Custom analog low power design: the problem of low voltage and mismatch. In *Custom Integrated Circuits Conference, 1997., Proceedings of the IEEE 1997*, pages 285–292, 1997.
- [23] M.J.M. Pelgrom, Aad C J Duinmaijer, and A.P.G. Welbers. Matching properties of mos transistors. *Solid-State Circuits, IEEE Journal of*, 24(5):1433–1439, 1989.
- [24] M. Conti, G.D. Betta, S. Orcioni, G. Soncini, C. Turchetti, and N. Zorzi. Test structure for mismatch characterization of mos transistors in subthreshold regime. In *Microelectronic Test Structures, 1997. ICMTS 1997. Proceedings. IEEE International Conference on*, pages 173–178, 1997.
- [25] B. Calhoun A. Wang and A. Chandrakasan. *Sub-Threshold Design for Ultra Low-Power Systems*. Springer, 2006.
- [26] D. Markovic, B. Nikolic, and R.W. Brodersen. Analysis and design of low-energy flip-flops. In *Low Power Electronics and Design, International Symposium*, pages 52 –55, 2001.
- [27] M. Alioto. Impact of nmos/pmos imbalance in ultra-low voltage cmos standard cells. In *Circuit Theory and Design (ECCTD), 2011 20th European Conference on*, pages 536 –539, Aug. 2011.
- [28] B.H. Calhoun, A. Wang, and A. Chandrakasan. Modeling and sizing for minimum energy operation in subthreshold circuits. *Solid-State Circuits, IEEE Journal of*, 40(9):1778–1786, 2005.
- [29] F. Klass. Semi-dynamic and dynamic flip-flops with embedded logic. In *VLSI Circuits, 1998. Digest of Technical Papers. 1998 Symposium on*, pages 108–109, 1998.
- [30] B. Nikolic, V. Stojanovic, V.G. Oklobdzija, Wenyan Jia, J. Chiu, and M. Leung. Sense amplifier-based flip-flop. In *Solid-State Circuits Conference, 1999. Digest of Technical Papers. ISSCC. 1999 IEEE International*, pages 282–283, 1999.
- [31] G. Gerosa, S. Gary, C. Dietz, Dac Pham, K. Hoover, J. Alvarez, H. Sanchez, P. Ippolito, Tai Ngo, S. Litch, J. Eno, J. Golab, N. Vanderschaaf, and J. Kahle. A 2.2 w, 80 mhz superscalar risc microprocessor. *Solid-State Circuits, IEEE Journal of*, 29(12):1440–1454, 1994.
- [32] H. Mostafa, M. Anis, and M. Elmasry. Comparative analysis of power yield improvement under process variation of sub-threshold flip-flops. In *Circuits and Systems (ISCAS), Proceedings of 2010 IEEE International Symposium on*, pages 1739–1742, 2010.
- [33] Consoli E. Alioto, M. and G. Palumbo. From energy-delay metrics to constraints on the design of digital circuits. *International Journal of Circuit Theory and Applications*, 40(8):815–834, 2012.

- [34] Xiaoying Yu, Xiaoyan Luo, and Jianping Hu. Low voltage and low leakage flip-flops based on transmission gate in nanometer cmos processes. In *Circuits and Systems (MWSCAS), 2011 IEEE 54th International Midwest Symposium on*, pages 1–4, 2011.
- [35] Bo Fu and P. Ampadu. Comparative analysis of ultra-low voltage flip-flops for energy efficiency. In *Circuits and Systems, 2007. ISCAS 2007. IEEE International Symposium on*, pages 1173–1176, 2007.
- [36] Meng Xiongfei, R. Saleh, and K. Arabi. Layout of decoupling capacitors in ip blocks for 90-nm cmos. *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, 16(11):1581–1588, 2008.
- [37] Tzu-Ming Wang, Ming-Dou Ker, and Sao-Chi Chen. Design of analog output buffer with level shifting function on glass substrate for panel application. *Display Technology, Journal of*, 5(9):368–375, 2009.